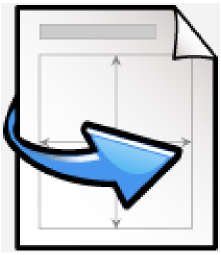
T³-Teachers Teaching with Technology™

TI-nspire™ CAS

Software version 3.6



Eksempelsamlingen 2. del: Grundlæggende færdigheder og begreber

0. Dokumentstyring
1. Variabelstyring

2. Ligninger og uligheder
3. Funktioner
4. Regressionsmodeller

5. Deskriptiv statistik
6. Sandsynlighedsregning
7. Hypotesetest

8. Vektorer og rumgeometri
9. Differentialligninger

TI-nspire CAS

Eksempelsamlingen 2. del — Grundlæggende færdigheder og begreber

Eksempelsamlingen vedligeholdes af
Mette Vedelsby, mette.vedelsby@skolekom.dk
Gert Schomacker, gert.schomacker@skolekom.dk
Bjørn Felsager, bjoern.felsager@skolekom.dk

Grundlæggende færdigheder og begreber er anden (og sidste del) del af **Eksempelsamlingen**. Den kan *ikke* bruges som en første introduktion til matematikprogrammet TI-*nspire*™ *CAS*, men forudsætter et vist kendskab til hvordan programmet er opbygget og hvordan man håndterer de enkelte værksteder. Som begynder anbefales man i stedet at starte med **Eksempelsamlingen første del: Introduktion til værkstederne**, eller man kan kigge i oversigtshæftet **Velkommen til TI-Nspire CAS**, der også indeholder aktiviteter til at komme i gang med TI-Nspire CAS.

I dette hæfte går vi mere i dybden med hvordan man arbejder med dokumenter i TI-Nspire CAS, herunder hvordan man kan skrive sammenhængende projekter i de to dokumentformater, der tilbydes: **DocumentView** og **PublishView**. Vi går også mere i dybden med hvordan de enkelte værksteder bindes sammen af **Variabelregistret**, der tillader information fra et værksted at styre hvad der sker i et andet værksted, og derved åbner mulighed for at man samtidigt kan arbejde med forskellige matematiske repræsentationer – tabeller, grafer og ligninger – i den matematiske modellering af et begreb eller et problem.

Derefter kommer der en række kapitler, der er emneorienterede, hvor vi kigger nærmere på grundlæggende matematiske teknikker og begreber. En gennemarbejdning af disse kapitler vil dels sætte læseren i stand til at håndtere rutineberegninger, såvel symbolsk som numerisk og grafisk, herunder selvfølgelig at løse simple eksamensopgaver. Men der vil også blive givet eksempler på hvordan man kan lege med de matematiske begreber og opbygge interaktive dynamiske dokumenter med værksteder, der spiller sammen og belyser forskellige aspekter af de matematiske begreber. Nogle af disse afsnit vil være mere udfordrende og det er ikke alle, der vil være motiverede for og have tålmodighed til at gennemarbejde sådanne projekter. Afsnit af en sådan mere udfordrende karakter vil være markeret med en * og kan overspringes ved en første gennemlæsning uden at man mister muligheden for at arbejde med de øvrige afsnit.

Man kan også sagtens uden større besvær læse kapitlerne i en vilkårlig rækkefølge, afhængig af på hvilket niveau og i hvilken rækkefølge de forskellige matematiske emner dukker op i undervisningen. Afhængigt af hvilken status lommeregneren har, vil man også kunne gøre **Beregnings**-værkstedet til det centrale regneværksted i stedet for som her **Note**-værkstedet. Kommandoerne er fælles og det skulle ikke være svært at styre eksemplerne fra et **Beregnings**-værksted i stedet.

Denne eksempelsamling har fokus på programmets anvendelse på computere, hvad enten der er tale om PC med Windows eller MAC-computere. Der er nogle ganske få forskelle på hvordan PC-versionen og MAC-versionen fungerer. Disse vil blive påpeget undervejs.

Programmet kan også anvendes på lommeregnerne/håndholdte og iPads. Måden man arbejder med programmet på en lommeregner eller en iPad adskiller sig dog så meget fra computerversionen, at det foreliggende hæfte ikke vil være hensigtsmæssigt til disse platforme.

Indhold

0 Dokumentstyring

I TI-Nspire CAS til computere kan du arbejde med to forskellige dokumentformater: **DocumentView** og **PublishView**. Begge er forberedt til at blive udskrevet på papir, men det første arbejder med halve ark, det andet med hele ark. De kan – med lidt forsigtighed – konverteres indbyrdes til hinanden.

DocumentView

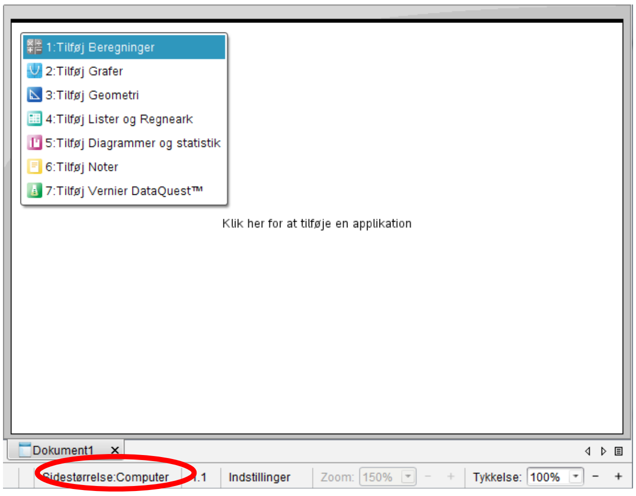
Obs
Husk at skifte til **Sidestørrelse Computer**, fx med **Ctrl/Cmd Shift Alt N**.

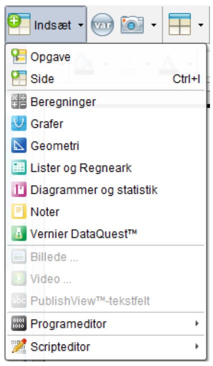
Vi vil først se på hvordan man hensigtsmæssigt kan arbejde med **DocumentView**.

Et klassisk TI-Nspire CAS dokument (tns) består af en række opgaver, der igen består af en række sider, der igen kan opdeles i et antal værksteder (op til fire).

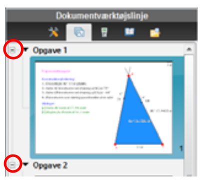
Tip
Ctrl/Cmd I er en meget nyttig genvej til indsættelse af nye sider.

Når du starter TI-Nspire CAS vil et nyt dokument blive oprettet. Dette dokument indeholder én opgave (Opgave 1) med én side, som skal have tilføjet ét værksted. Med **Indsæt** - knappen indsætter du flere sider i den aktuelle opgave eller en helt ny opgave. Her kan du også vælge mellem to programmeringsværksteder, idet TI-Nspire CAS understøtter to forskellige programmeringssprog: Et primitivt procedureorienteret programmeringssprog BASIC (**Programeditor**)og et avanceret objektorienteret scriptbaseret programmeringssprog LUA (**Scripteditor**). I øvrigt er TI-Nspire CAS selv skrevet i LUA, ligesom mange computerspil ☺. Vi kommer *ikke* ind på programmering i dette hæfte.





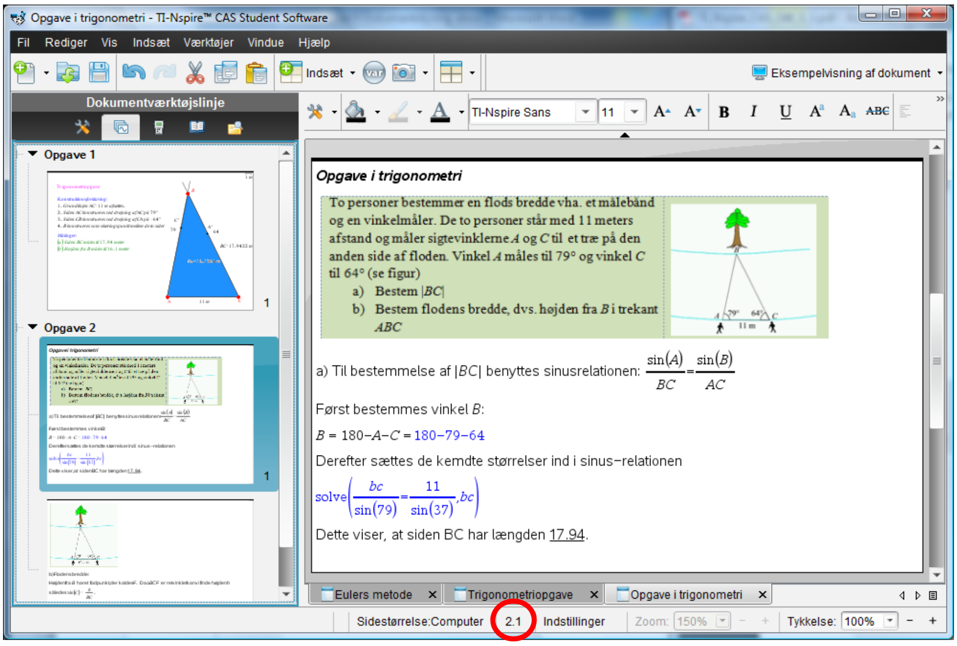
Tip
Du flytter rundt på siderne i **Sidesortereren** ved blot at trække til den ønskede placering.



På skærmbilledet næste side ser du **2.1** nederst i vinduet. Dette viser, at du aktuelt befinder dig i **Opgave 2, side 1**.

I sidepanelet til venstre vises siderne i det aktuelle dokument som miniaturer. Dette gør det bekvemt at bladre i dokumentet: Klik på en af miniaturerne og brug piletasterne til at bladre op og ned. Du kan dog også bruge rullebjælken til højre til at bladre i siderne, ligesom du kan bruge genvejen **Ctrl/Cmd ▸** (næste side) og **Ctrl/Cmd ◂** (forrige side).

Ved at klikke på et indfoldnings-ikon ⊟ henholdsvis et udfoldnings-ikon ⊞ åbner og lukker du for visningen af en **Opgave**. I miniature visningen kan du også ændre på rækkefølgen af siderne: Naviger til den side, du vil flytte, og træk siden hen, hvor du vil have den. Du kan også flytte sider mellem to opgaver, men det kræver, at der ikke er sammenfald af benyttede variabelnavne. Endvidere kan du slette den aktuelle side i miniaturevisningen med **DEL** — en handling du naturligvis kan fortryde med .



Tip
Du kan ændre opgavetitlerne ved at højreklikke på en titel og vælge **Omdøb F2** .

Variabelhåndtering

Det er vigtigt at forstå, hvordan TI-Nspire CAS håndterer variabeldeling: Alle variable, du opretter inden for en opgave, deles af alle de værksteder opgaven har fået tilføjet — uanset i hvilket værksted variablen er oprettet. Ændrer du en variabel i et af værkstederne, vil denne ændring automatisk slå igennem i alle værksteder, både fremadrettet og bagudrettet – the past is not protected!

Ved ændring af en variabel vil alle værkstederne altså blive opdateret med undtagelse af det beskyttede værksted **Beregninger**, hvis historik er upåvirket af ændringer i andre værksteder. Men genberegner du en variabel i **Beregninger**, der er ændret i et andet værksted, så vil du naturligvis se ændringen.

Opretter du en ny opgave, vil ingen variable fra tidligere opgaver være tilgængelige.

Sidelayout

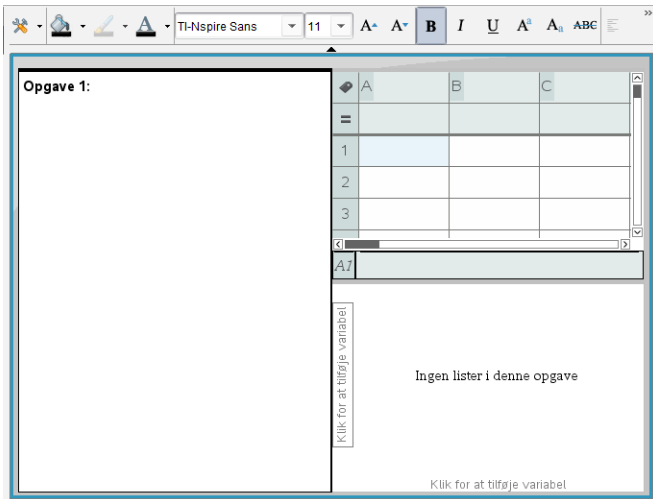
Spørgsmålet er nu hvordan vi mest hensigtsmæssigt sætter en opgavebesvarelse op, som udnytter TI-Nspire CAS mulighed for at anvende flere værksteder på én gang. Til enhver opgave hører der ledsagende tekst med forklaringer, uddybninger, konklusioner osv. Der er to værksteder hvor man kan blande tekst og beregninger: **Beregnings**-værkstedet og **Note**værkstedet.

Beregnings-værkstedet er glimrende til kladdeudregninger, hvor man skyder sig ind på problemet og forhåbentligt finder en løsning. Ydermere er det et beskyttet værksted, hvor de udregninger man har foretaget altid forbliver uberørte, uanset hvad man foretager sig i andre værksteder. Men det er meget klodset at rette i **Beregnings**-værkstedet og dermed også at tilrette tekst og beregninger, og man kan tilmed ikke formatere teksten.

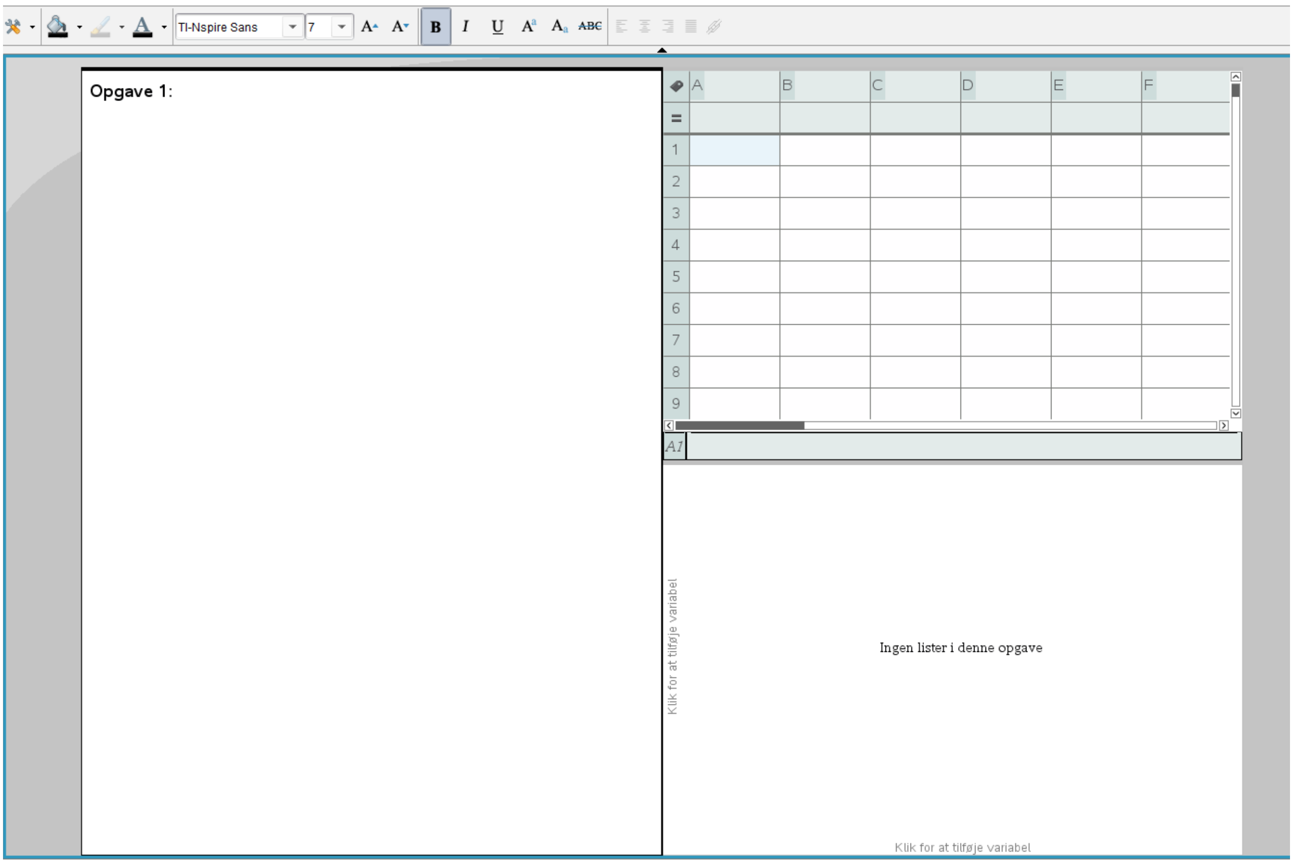
Note-værkstedet er et åbent dynamisk interaktivt værksted, men man kan dog godt beskytte udvalgte beregninger. Og så tillader det vidtgående formatering af teksten. Ydermere kan man sætte billeder ind i **Noter**, enten for at illustrere noget eller for at dokumentere noget i form af fx skærmklip fra andre værksteder. Enhver seriøs opgave besvarelse bør derfor have **Note**-værkstedet som udgangspunkt. Men da man også skal have andre værksteder i spil bør man overveje at bruge flere værksteder på samme side. En sådan opdeling i ruder sker nemmest med **Sidelayout**-værktøjet.

0: Dokumentstyring

En standardopsætning kan da fx se således ud, hvor man arbejder i to kolonner: I venstre kolonne har man *altid* et **Note**-værksted. I højre kolonner indsætter man værksteder efter behov. Her har vi vist det med et **Lister og regneark**-værksted og et **Diagrammer og statistik**-værksted som eksempel:



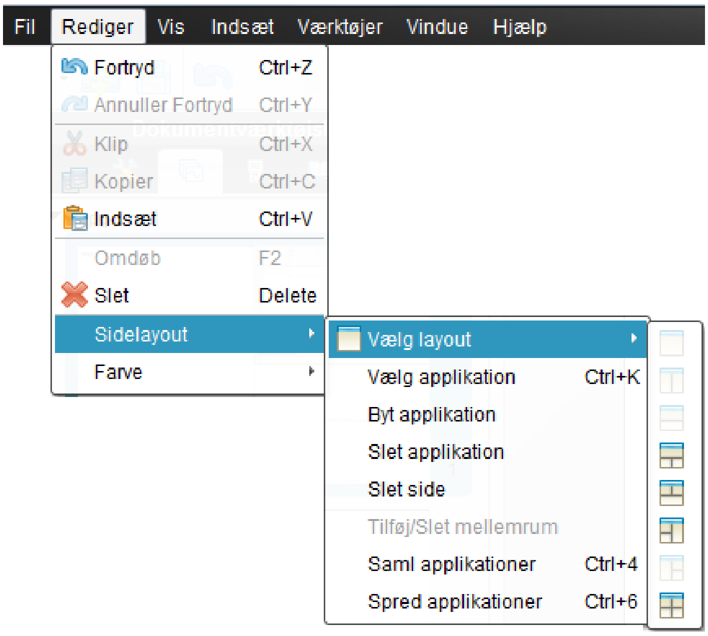
Arbejder du på en computer med stor skærm og høj opløsning kan du med fordel blæse programmet op til fuld skærm. I så fald skalerer **Note**-værkstedet med op og standard skriftstørrelsen på 11 pkt bliver for stor. Den bør da sættes ned til fx 7 pkt. Vi får da langt bedre udnyttelse af især regnearket, som nu får næsten samme plads at boltre sig på som i standardvinduet for **Sidestørrelse computer**. Men husk at dette dokument nu ikke uden videre kan overføres til en anden computer med lavere opløsning uden at læsbarheden i **Note**-værkstedet og antallet af rækker og søjler i **Lister og regneark**-værkstedet risikerer at gå tabt.



0: Dokumentstyring

Håndtering af værkstedsruder

Når man arbejder med mange værksteder i ruder på samme side rammer man uværgerligt ind i problemer omkring håndteringen af værkstedsruder, såsom hvordan man sletter en værkstedsrude, hvordan man bytter om på to værkstedsruder, hvordan man kopierer en værkstedsrude osv. Det håndteres alt sammen ved hjælp af sidelayout-menuen under **Rediger**-menuen på menubjælken foroven.



Mens det er nemt at håndtere siderne i dokumentet ved hjælp af **Sidesortereren**, så er det i **Sidelayout**-menuen de afgørende værktøjer til håndtering af værkstedsruder befinder sig ☺. Først og fremmest skal man kunne slippe af med en værkstedsrude eller kunne kopierere den over på en anden side. Det kræver at man først vælger ruden, dvs. *markerer* den.

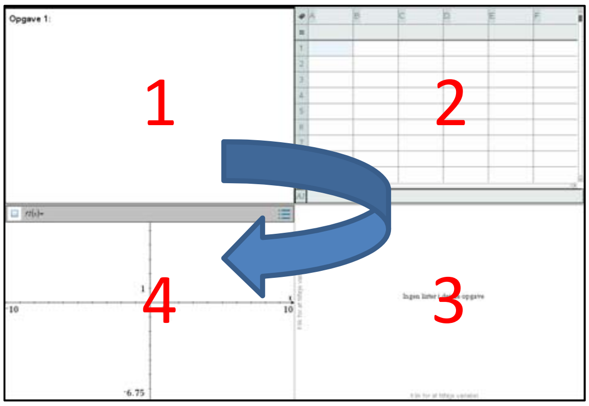
- Først skal ruden *aktiveres* ved at man klikker i ruden (hvorved rammen omkring ruden træder frem).
- Dernæst markeres den ved enten at benytte menupunktet **Vælg applikation**, bruge genvejen **Ctrl/Cmd K** (for *Kontrol af ruden*), eller ved simpelthen at dobbeltklikke på bjælken i toppen af ruden. Hvilken metode man end anvender kan man se ruden er markeret ved at rammen omkring ruden nu står og blinker.
- Endelig kan man nu slette ruden ved at taste **DEL** eller kopiere ruden ved at taste **Ctrl/Cmd C** osv. Derved lægges ruden på klippebordet og kan sættes ind på en anden side fx med **Ctrl/Cmd V**.(eller mere sikkert: **Kopier** og **Indsæt** fra **Rediger**-menuen).

Læg mærke til, at man kan ombytte to værkstedesruder ved at aktivere den ene værkstedsrude, vælge **Byt applikation** i **Sidelayout**-menuen og dernæst klikke i den anden værkstedsrude med ombytningsmarkøren .

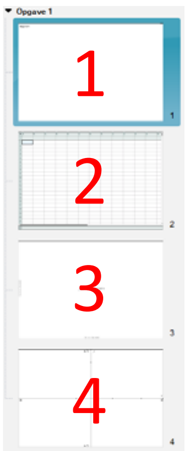
Endelig kan man sprede værkstedsruderne over flere sider ved hjælp af genvejen **Ctrl/Cmd 6** og samle dem igen på en enkelt side ved successivt at hente dem ind med genvejen **Ctrl/Cmd 4**. Fx er det nemmere at arbejde i et **Lister og regneark**-værksted, hvis det får lov til at udfylde hele siden. Når man er færdig kan de andre ruder så hentes tilbage igen. Tilsvarende er det nemmere at udføre detaljerede geometriske konstruktioner, hvis **Geometri**-værkstedet får lov til at fylde hele siden. Når man er færdig kan de andre ruder så hentes tilbage igen.

0: Dokumentstyring

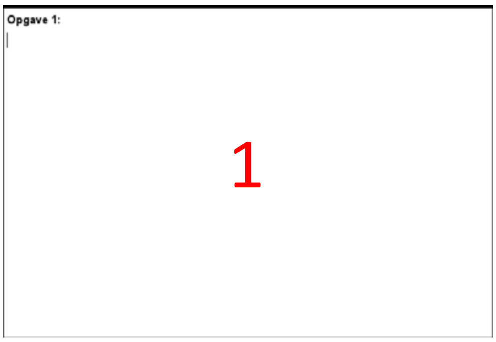
Når man samler eller spreder værktstederne kan det betale sig at være opmærksom rudernes rækkefølge:



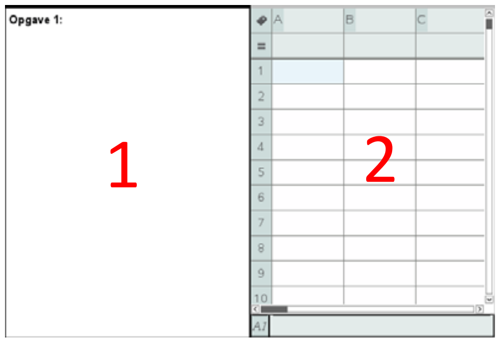
Ruderne går altså med uret rundt fra øverste venstre hjørne til nederste venstre hjørne. Hvis man spreder ruderne med menupunktet **Spred applikationer**, dvs. genvejen **Ctrl/Cmd 6,** dukker de altså op som fire særskilte sider i den ovennævnte rækkefølge. Samler vi dem igen – en efter en – med menupuntket **Saml Applikationer**, dvs. genvejen **Ctrl/Cmd 4** hentes de selvfølgelig ind igen i den samme rækkefølge:



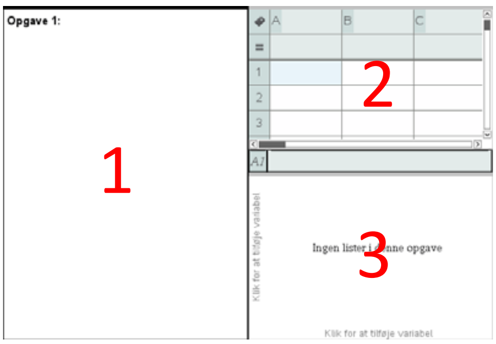
Sidesortereren efter værksteder-ne er spredt på hver sin side med **Ctrl/Cmd 6**.



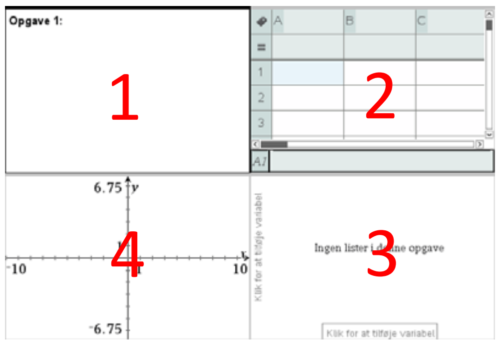
Udgangssiden



Første gang **Ctrl/Cmd 4**



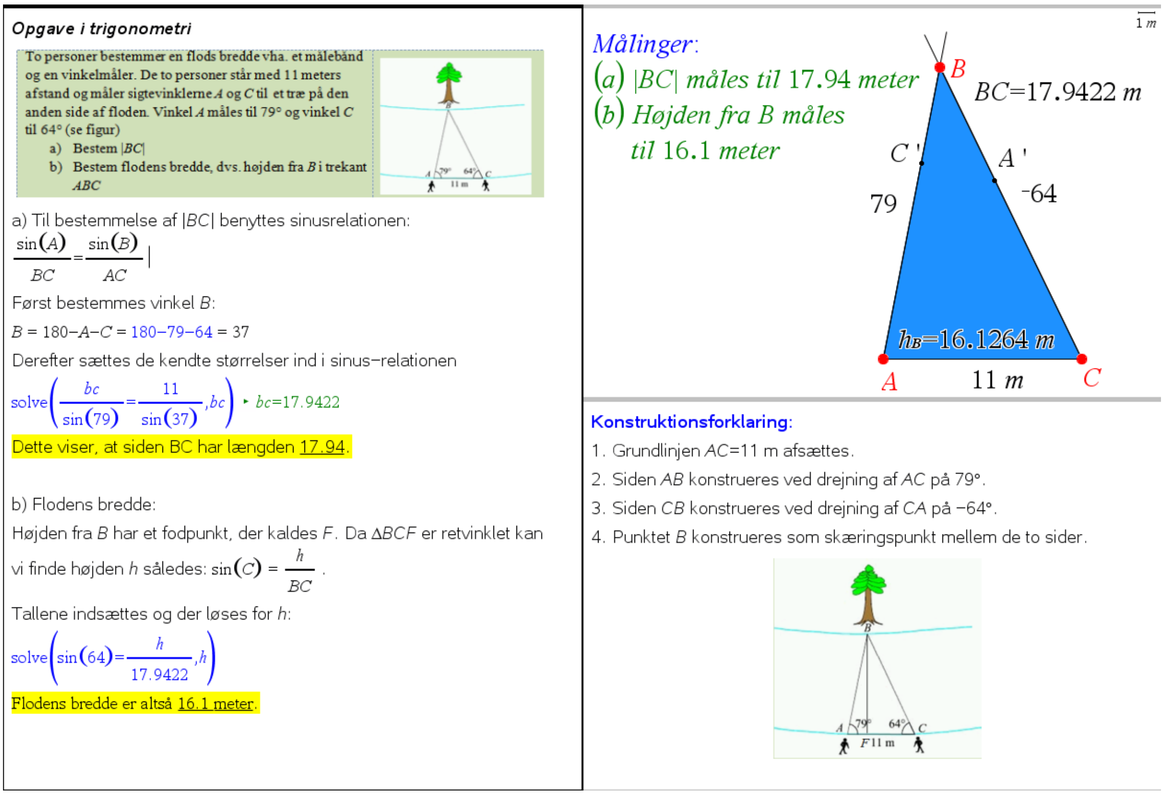
Anden gang **Ctrl/Cmd 4**



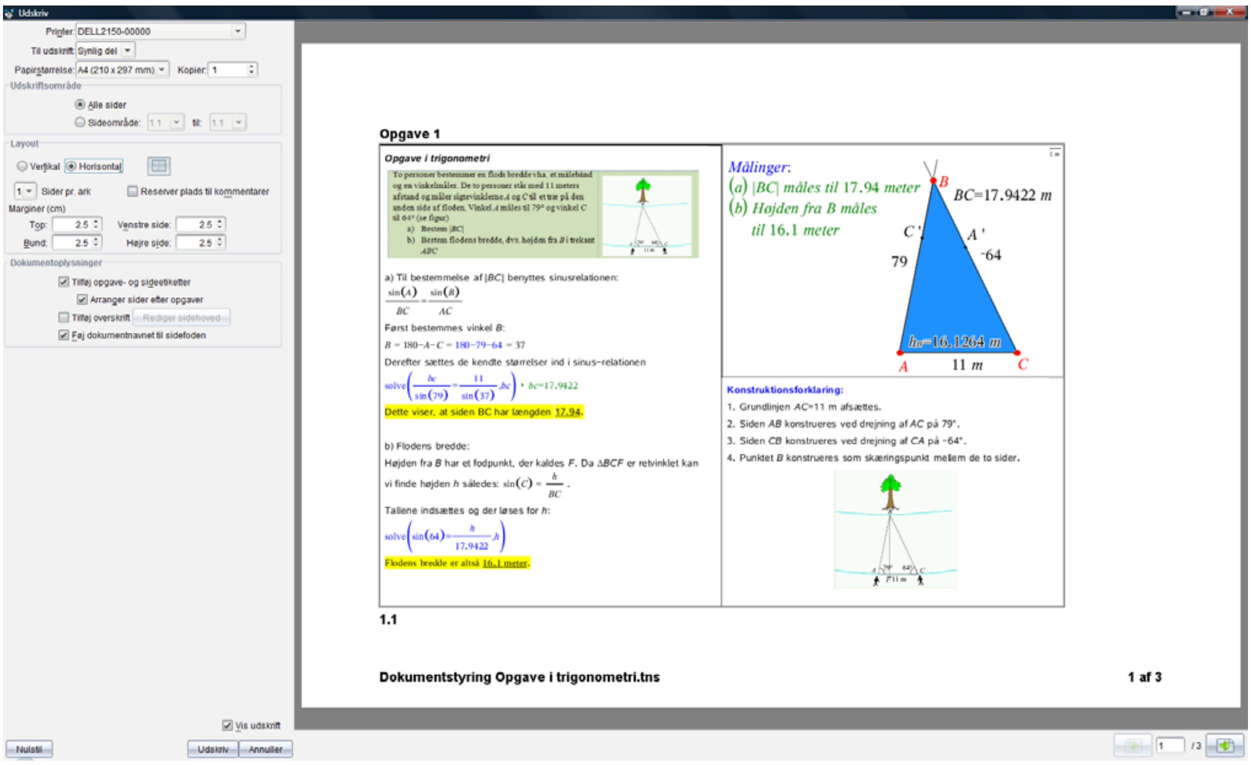
Sidste gang **Ctrl/Cmd 4**

Vi vender nu tilbage til trigonometriopgaven fra starten af kapitlet og viser hvordan den ser ud i en standardopsætning i **DocumentView**. Vi sætter den altså op i to spalter på en stor skærm. Venstre spalte er en noterude med opgaveteksten sat ind som billede og derefter kommenterede udregninger med konklusioner på spørgsmålene. Højre spalte er først en geometrirude med konstruktionen af trekanten og de tilhørende måling. Derefter kommer endnu en note-rude med konstruktionsforklaringen samt en skitse (tillempet fra opgaveteksten), der viser højden fra *B*. Ved udmålingen af højden behøver vi ikke konstruere højden, da vi bare skal måle afstanden fra punktet *B* til grundlinjen *AC*. Hele opgaven kan altså indpasses på en enkelt side.

0: Dokumentstyring



Men dermed kan opgaven også skrives ud på en enkelt side.

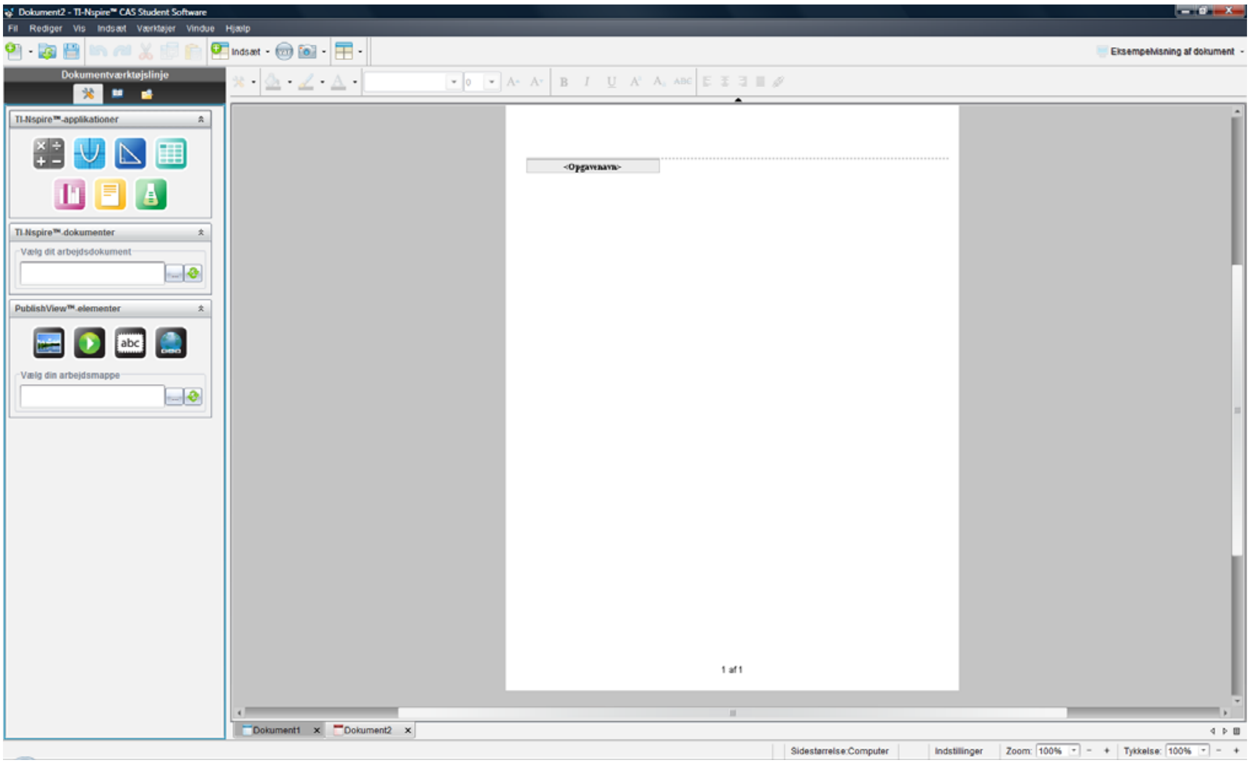


Vi skal blot huske at sætte **Antal sider pr ark** til 1 og **Layout** til **Horisontal** (Landscape).

PublishView

Obs
Du kan ikke sende et **PublishView** dokument til en håndholdt. En **PublishView** fil kan dog konverteres til en **tns fil**, der kan overføres.

Vi vender os nu mod alternativet til **DocumentView**, det såkaldte **PublishView**. Som navnet antyder, er det målrettet rapportskrivning, der fra starten skrives som et dokument på papir. Åbner man for et **PublishView** dokument (fx med **Ctrl/Cmd Shift N**) dukker det første ark op på skærmen. Man kan zoome ind og ud på arket, fx for at få lov til at se hele arket på en gang. På en computer med god skærmopløsning kan man dog godt se hele arket på en gang uden at zoome.



Sidestørrelsen

Man skulle nu tro man bare kunne begynde at skrive løs på arket. Men det kan man ikke! Alt hvad man laver på arket foregår i en værkstedsrude, der *trækkes* ind fra sidepanelet.

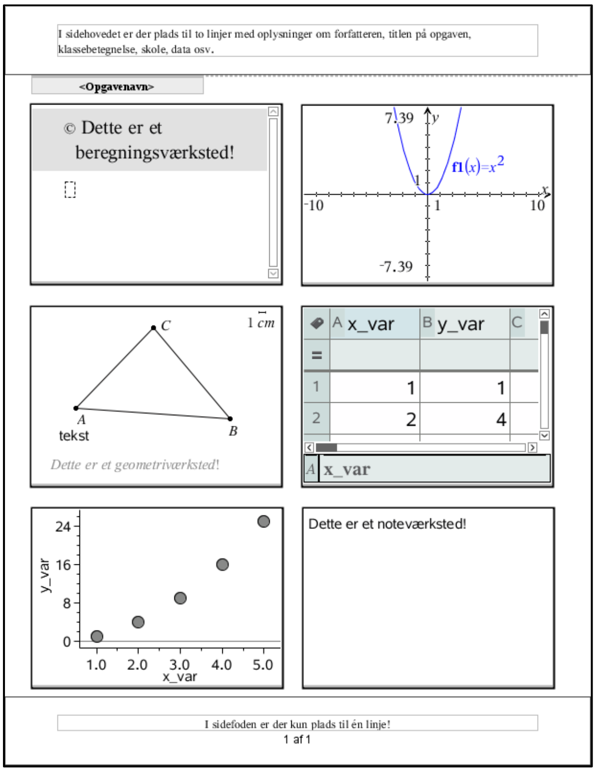
Man skulle også umiddelbart tro at man arbejder på et A4-ark. Men det er ikke tilfældet. Arkets format følger det amerikanske Letter-format, som er en anelse bredere, men til gengæld ikke så højt:

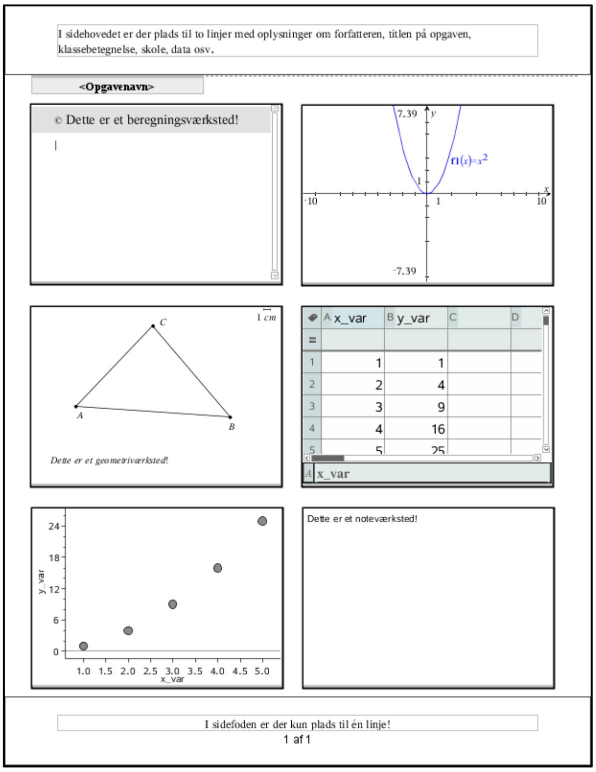
A4-ark = 21.0 cm × 29.7 cm
Letter-ark = 21.6 cm × 27.9 cm

Når man skriver ud mister man altså en ca. halv cm i bredden, hvorfor højre margen afstumpes lidt. Til gengæld får man godt 2 cm mere i højden, som blot bliver til spildplads, da der samtidigt er sat god plads af til sidehoved og sidefod. Skrivefeltet er derfor reduceret til

Skrivefelt = 20.0 cm × 22.7 cm

Der er derfor *ikke* plads til 2 store værkstedsruder i sidestørrelse computer, hvis de sættes i fuld bredde. Nu er det sådan at værkstedsruderne som udgangspunkt hentes ind i Sidestørrelse håndholdt. Da Sidestørrelse håndholdt kun er halvt så stor som sidestørrelse computer er der derfor pæn plads til 6 værkstedsruder i standardstørrelse:



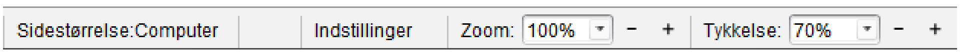


Tykkelse 100% **Tykkelse 70%**

(Tykkelse minder om zoom, men fx bliver geometriske figurer ikke større, de bliver bare tykkere ☺)

Obs
Det er meget tungere for computeren at arbejde i **PublishView**, så det egner sig ikke til små computere med begrænset arbejdshukommelse ☹.

Som det ses, lider især **Beregnings**-værkstedet og **Lister og Regneark**-værkstedet under at blive vist i **'Sidestørrelse håndholdt'**. Det kan derfor kraftigt anbefales at man fra starten sætter skaleringen ned for alle værkstederne. Det sker i den såkaldte **Tykkelse** (linje-tykkelse) i statuspanelet forneden.



Sættes den ned til fx 70% ser det straks mere acceptabelt ud ☺. Og husk at vi stadigvæk arbejder med standardruder, der kun er halvt så store som de værkstedssider vi benytter i **DocumentView**.

Fast sidelayout

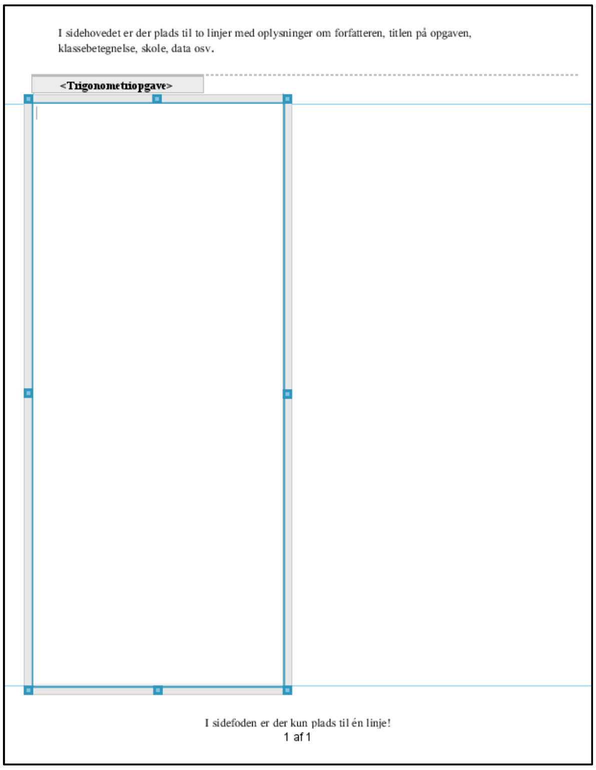
Spørgsmålet er nu igen hvordan man mest hensigtsmæssigt sætter en opgavebevarelse op. Det afhænger til en vis grad af ens erfaring. Til at begynde med er det nemmere at styre et fast men knap så fleksibelt format. Senere hen kan man så slå sig mere løs ☺

Vi lægger derfor ud med et standardformat, hvor vi skriver i to søjler: En venstre søjle til **Note**-værkstedet og en højre søjle til de supplerende værksteder, der måtte blive brug for hen ad vejen.

Når man trækker **Note**-værkstedet ind i arket kan man flytte det op under opgaveikonet og bruge det som 'anker' for at få det lagt på plads med samme venstre margen som **Opgavetitlen**. Man kan nemt se, hvis man trækker det for højt op ☺. Når først **Note**-værkstedet er sluppet kan man gribe fat i ankeret forneden og trække den nederste kant helt ned til bunden af arket:

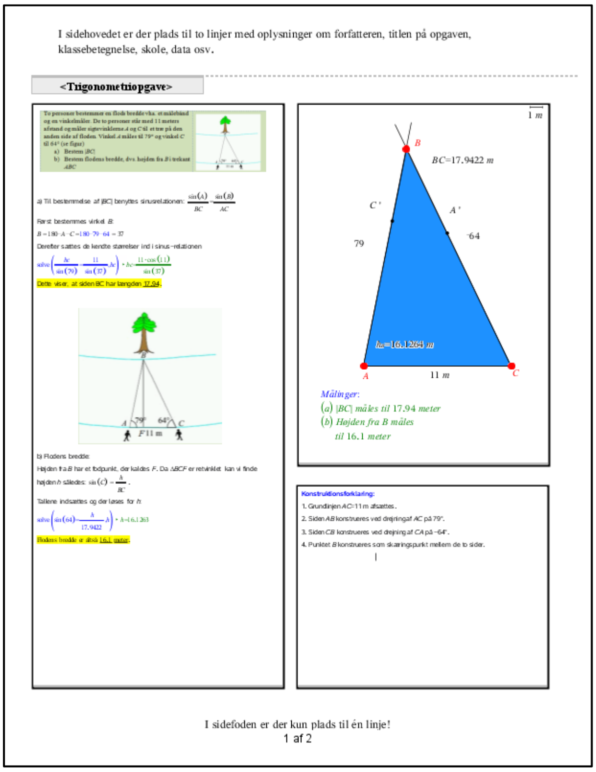
0: Dokumentstyring

Tip
Du ændrer opgavens titel ved at skrive løs i titelfeltet ☺



Læg mærke til de blå formatteringslinjer, der dukker op, når man trækker i ankrene. De kan bruges til at sætte værkstedruden præcis i forhold til andre objekter på arket.

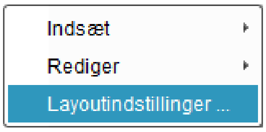
Der er nu rigelig plads til at begynde at skrive løs og sætte den samme opgave op som før. Her har vi trukket et **Geometri**-værksted ind i højre søjle og givet det god plads i højden. Konstruktionsforklaringen har vi paceret nedenunder i en selvstændig **Note**-rude.

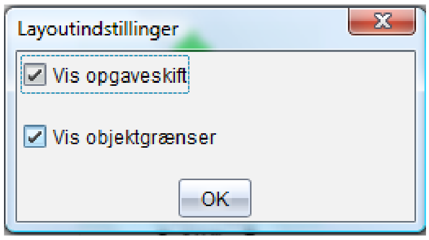


0: Dokumentstyring

Det kan godt være du synes det ser anmassende ud med rammer omkring værksteds-ruderne, men dem kan man fjerne igen ved at højreklikke i arket og vælge **Layoutindstillinger**:

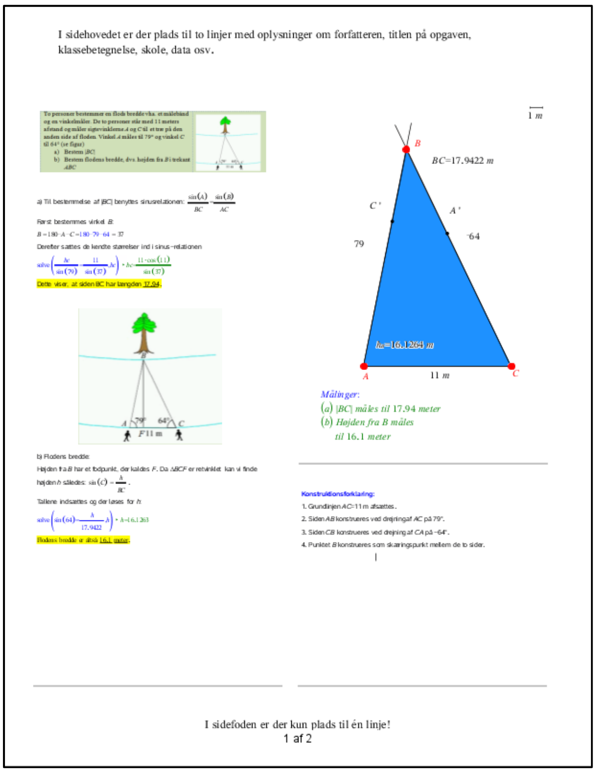
Tip
Selv om du bevarer rammerne omkring værkstedsruderne kan du godt fjerne dem i udskrivningen af dokumentet. Det samme gælder opgaveskiftene ☺





Fjerner vi hakkene ser det således ud:

Tip
Det er nemt at markere en rude ved at klikke på rammen og flytte rundt med den – også til helt andre ark. Det er også nemt at slette ruden eller kopiere den fra et dokument til et andet med **Ctrl/Cmd C** efterfulgt af **Ctrl /Cmd V.**



Tip
Selv om du i princippet kan opdele en værksstedsrude i underruder er det overflødigt i **PublishView** ☺

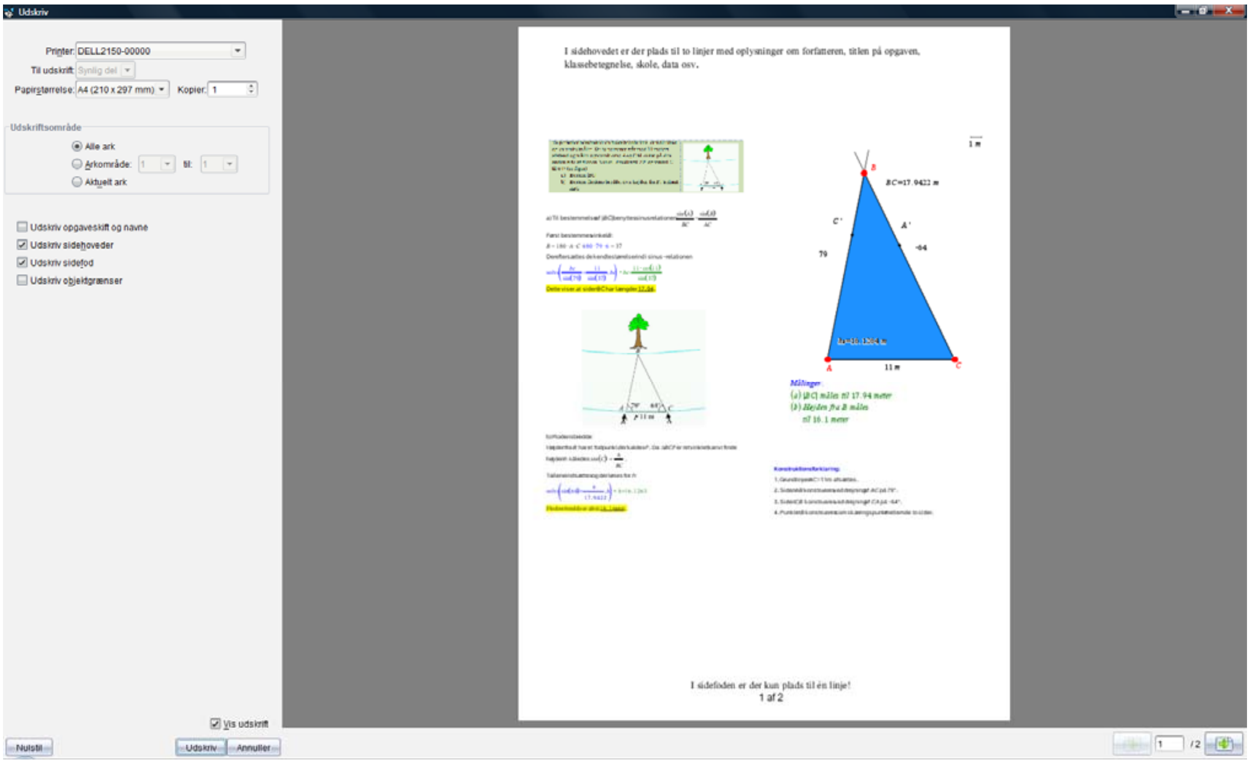
Når man ikke kan se opgaveskiftene kan man altså heller ikke længere se opgavenavnet! Det kan så i stedet tilføjes i **Note**-værkstedet ☺.

Tip
Du fjerne et opgaveskift ved at klikke i opgavetitellinjen og derefter klikke i det røde kryds ✕.

Du kan dog også tilføje rammerne og opgavetitlerne i forbindelse med udskrivningen. Her åbner du bare for **Print**-dialogboksen og sætter kryds de rigtige steder. Da **Print**-vinduet indeholder en forhåndsvisning af udskrivningen, kan du faktisk bladre hele dokumentet igennem og læse det på skærmen for at se om udskrivningen bliver som forventet. Læg mærke til luften omkring sidehovedet og sidefoden. Det skyldes som sagt, at vi udskriver til et A4-ark fra et Letter-ark ☺

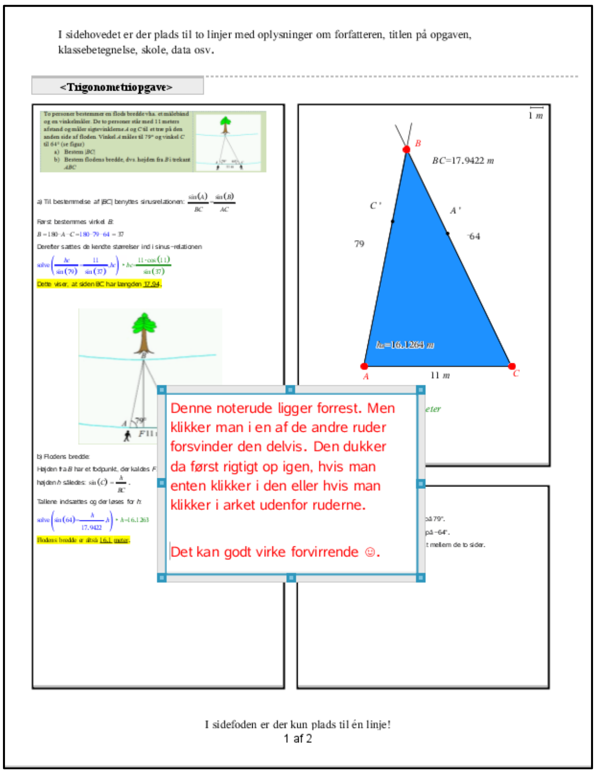
Det faste sidelayout er meget nemt at håndtere. Når der ikke er plads mere på et ark indsætter vi blot et nyt ark (gerne med **Ctrl/Cmd I**). Her sætter vi igen en **Note**-rude i venstre søjle og fortsætter … .

Har vi brug for en ny opgave, indsætter vi bare et nyt ark og på det nye ark starter vi med at indsætte en ny opgave, der vises som et opgaveskift i toppen. Men der er ikke meget spræl i det faste sidelayout ☺.

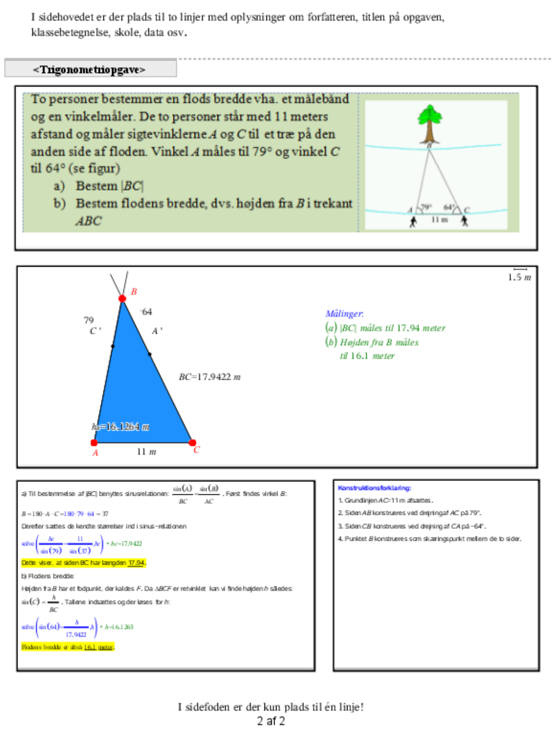


Løst sidelayout

Man kan godt få mere fleksibilitet ind i layouten ved at undlade at arbejde i søjler. Man kan da sætter ruderne præcis som man vil. Igen kan man bruge de blå forankringslinjer til at trække ruden på plads i den ønskede placering i forhold til de andre objekter på arket. Man kan endda lade ruderne overlappe og bestemme hvem der skal ligge øverst og hvem der skal ligge nederst, når ingen af ruderne er aktive (for når de aktiveres rykker de altid i front). Men overlappende ruder kan godt virke forvirrende, så det bør man normalt holde sig fra.



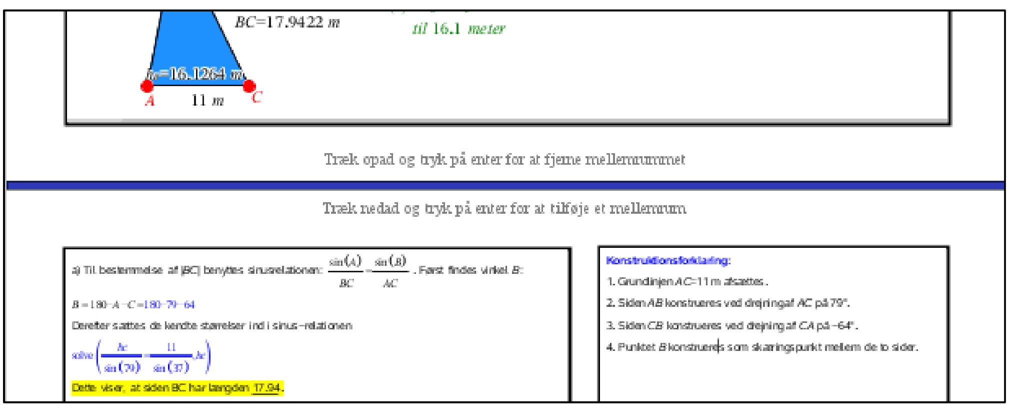
Man kan da bekvemt arbejde med ruder der strækker sig på tværs af hele arkets bredde. Trigonometriopgaven kan da fx se således ud:



Hvis man foretrækker et mere løst layout løber man dog hurtigt ind i det problem, at der skal skaffes plads på en side, fordi man vil udvide en beregning eller indsætte et ekstra billede osv. Det kan også være man fortryder et afsnit og pludselig står med en tomt gabende hul på en side.

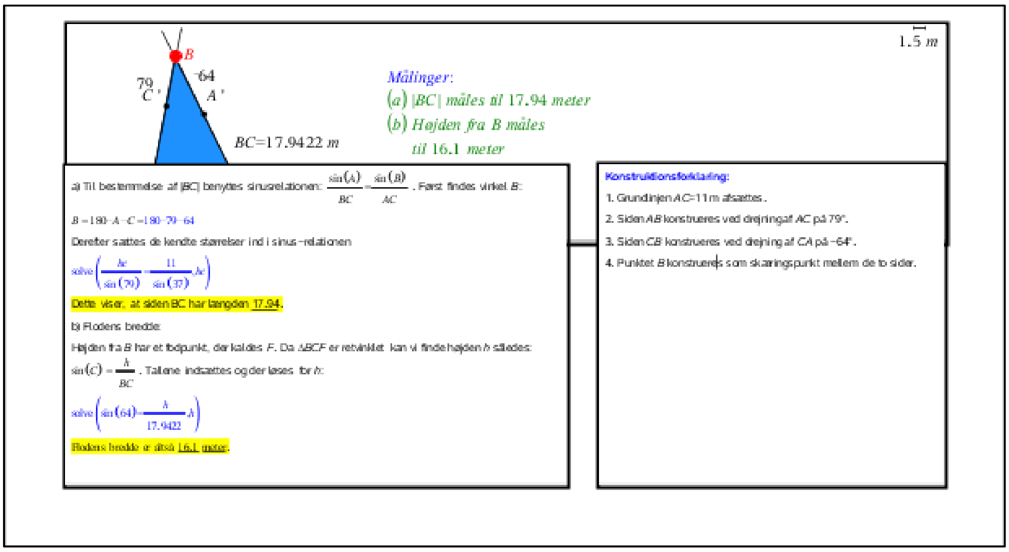
Man skal da højreklikke et eller andet sted i arket, hvor man gerne vil tilføje plads eller vil indskrænke pladsen. Derefter vælger man **Rediger** ▸ **Tilføj/Slet mellemrum**.

Der dukker da en skillelinje op som fæstnes ved at man klikker på den:

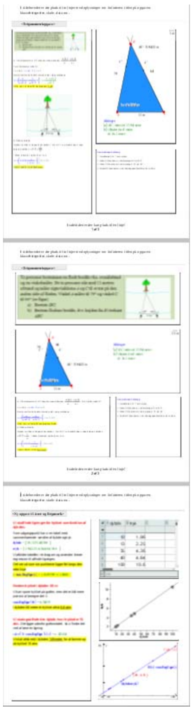


0: Dokumentstyring

Som det ses kan man nu enten trække opad (gerne med piletasten ▲) for at fjerne et rødt mellemrum, eller man kan trække nedad (gerne med piletasten ▼) for at skabe et grønt mellemrum:



Her er der trukket – for langt – opad ☺.

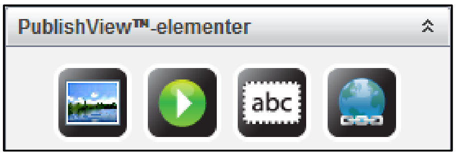


Med lidt øvelse kan man reparere på layouten på denne måde. Skubber man tilstrækkeligt meget nedad, ryger værkstedsruderne over på en ny side, der oprettes automatisk, hvis der ikke allerede er en.

Til sidst bemærker vi at der *ikke* findes nogen **Sidesorterer** i **PublishView**. Men man kan skaffe sig oversigt over siderne ved simpelthen at zoome kraftigt ud (her vist med 35%).

Specialværksteder i PublishView

Udover de traditionelle matematikværksteder kommer PublishView med sine egne fire nye medieværksteder, der kun virker i **PublishView**, dvs. de kan ikke overføres til tns-dokumenter (endsige håndholdte). To af medieværkstederne bibringer dog ikke noget egentligt nyt, så de kan reelt erstattes af matematikværksteder, der stort set kan det samme.

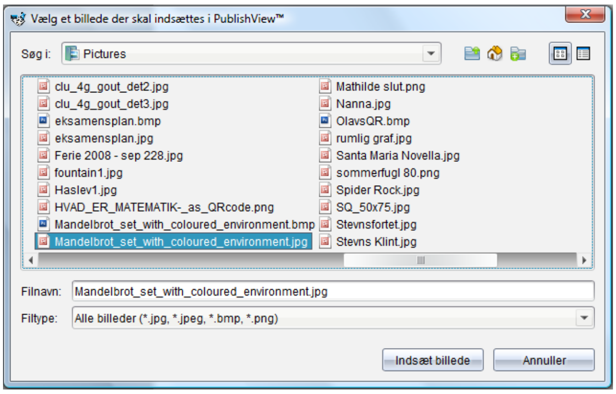


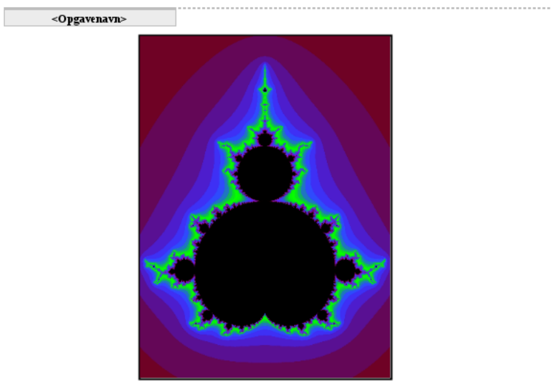
Der er tale om et **Billed**-værksted, et **Video**-værksted, et **Tekst**-værksted og et **Hyperlink**-værksted. Her er **Tekst**-værkstedet reelt dårligere end **Note**-værkstedet, så det er der faktisk ingen grund til at bruge krudt på. Da man også kan indsætte billeder i **Note**-værkstedet vil man under normale omstændigheder heller ikke have brug for **Billed**-værkstedet ☺.

Reelt kommer de nye mediemuligheder derfor fra **Video**-værkstedet og fra **hyperlink**-værkstedet. Vi ser nu kort på mulighederne hver for sig:

1. Billed-værkstedet:

Trækkes billedværkstedet ind på arket, åbnes en dialogboks med en stifinder, hvorfra man kan hente billedet i et af formaterne **jpg**, **bmp** og **png**:

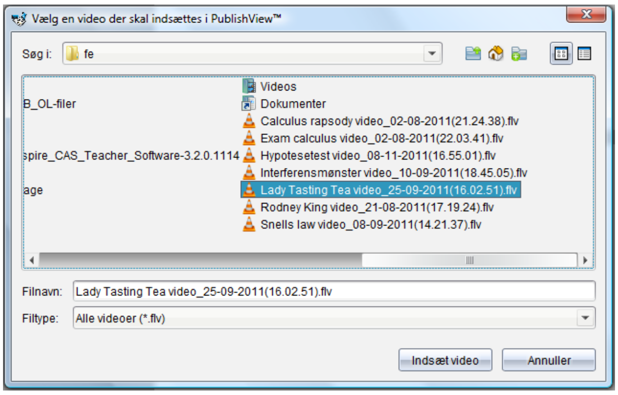




Men husk at man også kan sætte billeder ind i et **Note**-værkstedet, så det er kun specielle billeder i en særlig kvalitet, som bør sættes op med **Billed**-værkstedet, da disse billeder, jo *ikke* kan overføres til **DocumentView** og dermed heller ikke til håndholdte.

2. Video-værkstedet:

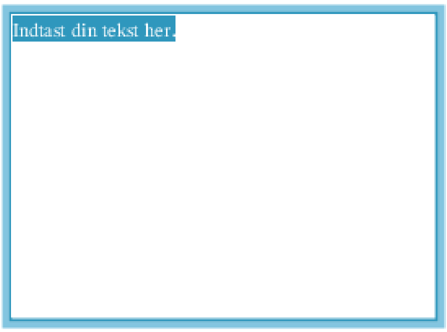
Trækkes videoværkstedet ind på arket, åbnes en dialogboks med en stifinder, hvorfra man kan hente videoer gemt i formatet **flv**:



Videoer er et stærkt medie til at introducere nye emner, herunder til instruktionsvideoer i brugen af TI-Nspire CAS ☺. Men de kan altså kun vises i **PublishView**-formatet.

3. Tekst-værkstedet:

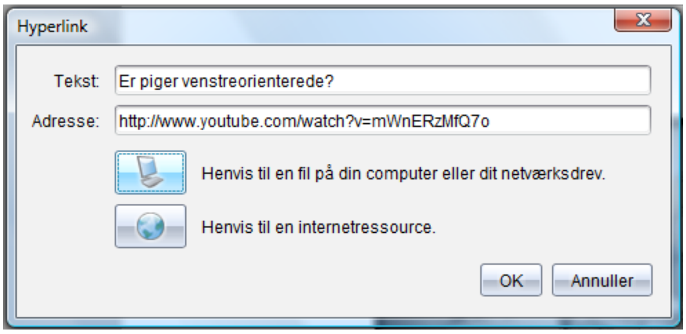
Trækkes tekstværkstedet ind på arket, åbnes en tekstboks som man kender den fra Word. Det åbner mulighed for simpel formatering af teksten:



Vi får dog ikke mulighed for at benytte hævet skrift, sænket skrift eller overstreget skrift, ligesom vi ikke kan indsætte skabeloner. I forhold til **Note**-værkstedets muligheder er det altså en pauver omgang, hvor det alene er muligheden for at justere linjerne (venstrejusteret, centreret osv.) samt muligheden for at indsætte et hyperlink der åbner nye muligheder. Men et hyperlink kan man jo lige godt indsætte med **Hyperlink**-værkstedet. Alt i alt er det altså svært at se fordelen med tekstværkstedet i dets nuværende udformning, og da det ydermere ikke kan overføres til **DocumentView** skal der tungtvejende grunde for at begynde at blande det ind i **PublishView** ☺.

4. Hyperlink-værkstedet:

Trækkes hyperlinkværkstedet ind på arket, åbnes en dialogboks, hvrofra man kan indsætte et hyperlink til en anden fil på computeren eller til et websted.



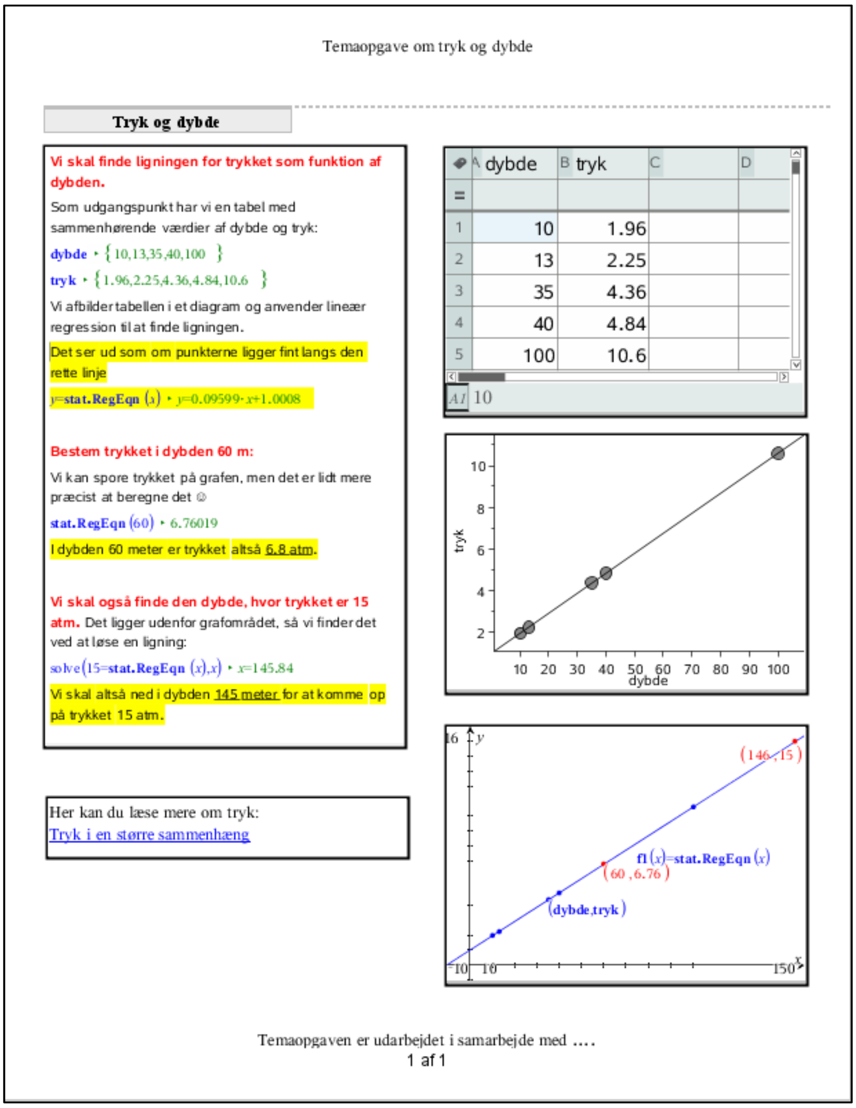
Som det ses, indsættes der en tekstboks med hyperlinkteksten skrevet i blå understregning, for at understrege at der er tale om et hyperlink. Man kan så efterfølgende som vist supplere med lidt forklarende tekst. Klikker man på linket åbner man i dette tilfælde for YouTube filmen **Er piger venstreorienterede?**

Hyperlinks er et stærkt værktøj til at henvise til fx en instruktiv YouTube-film, som den ovenstående, eller et uddybende dokument, fx et andet **PublishView** dokument, der går mere i dybden med en eller anden detalje.

Et eksempel på et PublishView dokument

Du kan nu øve dig i at opsætte et **PublishView** dokument med brug af mange af de omtalte færdigheder. Det følgende eksempel indeholder værkstederne: **Noter**, **Lister og Regneark**, **Diagrammer og Statistik**, **Grafer** samt **PublishView**-værkstedet **Hyperlink**.

Prøv selv at genskabe dokumentet nedenfor — ikke ved at kopiere gammelt arbejde til **PublishView**, men helt fra bunden af. Det er sådan du kommer til at arbejde med opgaveløsning i TI-Nspire CAS ☺



Den færdige opgave kan du udskrive med menuvalget **Filer ▸ Udskriv...**

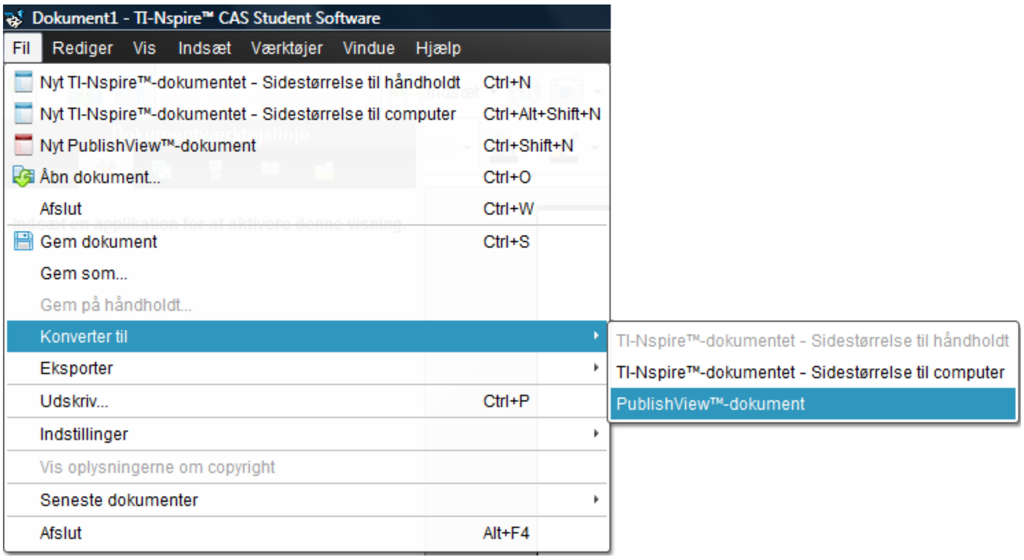
Samspillet mellem DocumentView og PublishView

Når man har arbejdet i **DocumentView**, fx fordi det er hurtigere at arbejde med i timerne på skolen, så ville det være rart efterfølgende at kunne konvertere **tns**-dokumenterne til **tnsp**-dokumenter, dvs. **PublicView**-dokumenter, fx fordi der skal udarbejdes en rapport. I så fald er der nogle specielle forhold man skal være særligt opmærksomme på.

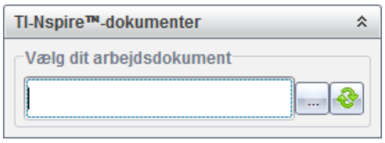
For det første vil man ofte arbejde med sideopdeling i ruder i **DocumentView**, så man har flere værksteder på samme side. Før man konverterer et **tns**-dokument, bør disse ruder derfor opløses og fordeles på særskilte sider. Det sker hurtigt ved at gå dokumentet igennem og taste **Ctrl/Cmd 6** på de sider, der skal opløses.

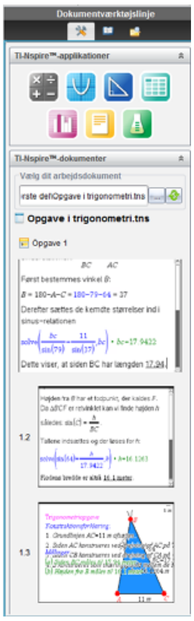
Dernæst bør man overveje om det kan betale sig at konvertere hele dokumentet i ét hug. Det kan man nemlig gøre ved hjælp af

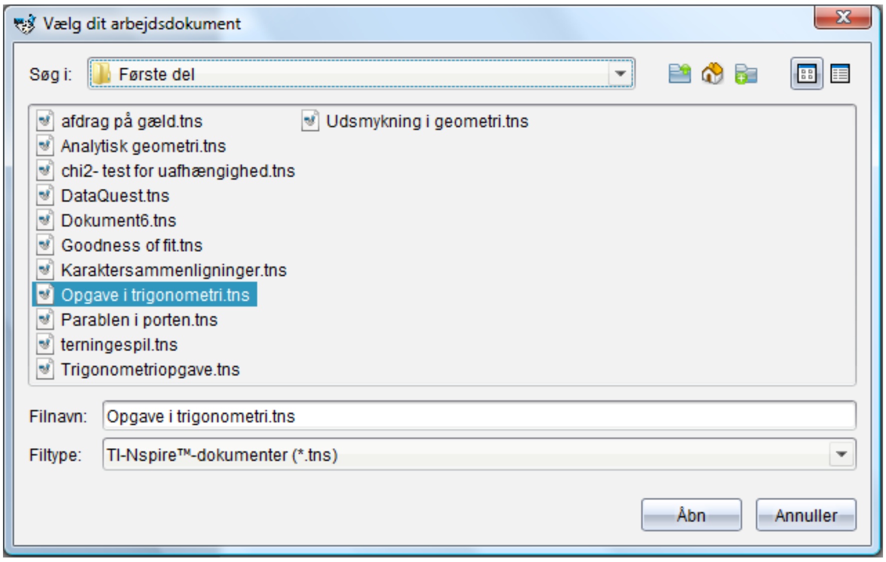
Filer ▸ Konverter til … ▸ PublishView



Man kan imidlertid også vælge at kontrollere processen selv, hvilket giver langt større fleksibilitet i hvad man vil have overført og hvordan. Man skal da åbne for stifinderen  i sidepanelet under **TI-Nspire™ dokumenter**, der giver adgang til ens **tns**-dokumenter:







Herefter kan man trække de sider fra tns-dokumenterne man har brug for, ind i den rækkefølge der nu måtte være passende.

1 Variabelstyring

I arbejdet med TI-Nspire CAS er det meget vigtigt at forstå, hvad variable er, hvordan de håndteres, samt at kende betydningen af *konkrete variable*, der har fået tilskrevet en værdi og *symbolske variable*, der endnu ikke har fået tilskrevet nogen værdi.

Variable kan dække over mange ting, men her vil vi først og fremmest fokusere på *talvariable*, dvs. variable hvis værdier er tal. Man kan tænke på en variabel som et spillekort, hvor variablens navn står på den ene side, dens værdi på den anden side. Så længe man kun spiller med navnesiden opad, håndteres variablen som en bogstavvariabel, der opfører sig på præcis samme måde som tal, når man regner på den, fx lægger den sammen med andre bogstavvariable. Men når variablen tildeles en værdi, dvs. man vender kortet, skifter den status og der regnes udelukkende videre på talværdien.

En egentlig symbolsk variabel er så et kort, hvor der godt nok står et navn på den ene side, men der er endnu ikke indskrevet nogen værdi på den anden side af kortet.

Toppunktsformlen for en parabel

I dette afsnit benyttes som eksempel toppunktsformlen for en parabel

$$T=\left(-\frac{b}{2a},-\frac{d}{4a}\right)$$

hvor a, b og c er koefficienterne i parablens ligning $y=a\cdot x^2+b\cdot x+c$, mens $d=b^2-4a\cdot c$ er diskriminanten hørende til parablens ligning.

Vi vil som udgangspunkt arbejde i **Note**-værkstedet, som er et interaktivt dynamisk værksted, hvor man for alvor kan se variablene udfolde sig. Det kan godt i starten virke lidt mere forvirrende end det beskyttede **Beregnings**-værksted, hvor udregninger ikke opdateres. Men det er afgjort umagen værd at vænne sig til at arbejde i **Note**-værkstedet ☺. Hold godt øje med hvad der sker med matematikfelterne undervejs, når du gennemarbejder det følgende eksempel ☺

Variablene optræder også i de øvrige værksteder, hvilket vi undervejs vil se mange eksempler på.

1. Tildeling af talværdier til variable

I det konkrete eksempel, $y=x^2+2x-1$, er $a=1$, $b=2$ og $c=-1$. På TI-Nspire CAS kan du gemme talværdier i *navngivne variable*:

Tip
Alternativt kan du lave tildelingen således $1 \rightarrow a$, hvor pilen findes i symbolpaletten - eller du kan bruge genvejen =: (lig med kolon), dvs. 1:=a. Endelig kan du bruge define-kommandoen, dvs. taste define $a = 1$

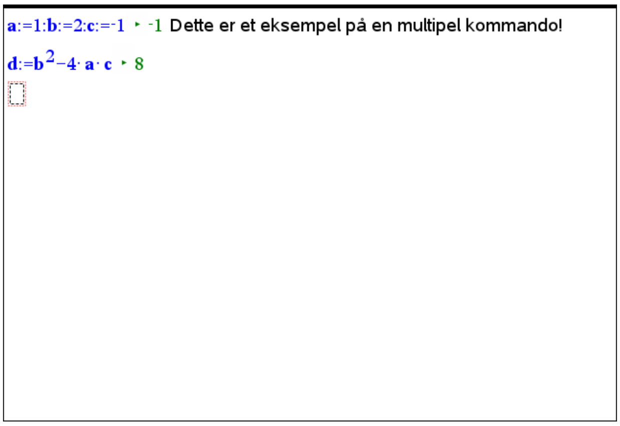
Et 1-tal gemmes i variablen a ved at taste: a:= 1 i et matematikfelt (**Ctrl/Cmd M**), og tilsvarende for de andre variable, se skærmbilledet næste side.

Du kan også gemme værdien af et udtryk/formel i en variabel, fx kan du gemme værdien af $b^2-4a\cdot c$ i en variabel med navnet d. Du skal blot taste $d:=b^2-4a\cdot c$.

Man siger at a, b og c er *uafhængige variable*, idet det står frit for at udskifte deres værdi som det måtte passe os.

Til gengæld er d en *afhængig variabel*, idet dens værdi afhænger af værdien af de frie uafhængige variable a, b og c. Vi kan derfor kun udskifte værdien for den afhængige variabel *d* indirekte ved at ændre på værdierne for de uafhængige variable *a*, *b* og *c*. Hvis vi forsøgte os med at tildele variablen *d* en talværdi direkte ved en kommando på formen $d := \ldots$ ville sammenhængen med variablene *a*, *b* og *c* blot blive brudt.

Tip
Du behøver ikke at taste **Enter** efter hver tildeling, men du må gerne. Det bliver mere overskueligt, hvis tildelingerne adskilles af et kolon og udføres som en samlet kommando indenfor et enkelt matematikfelt.



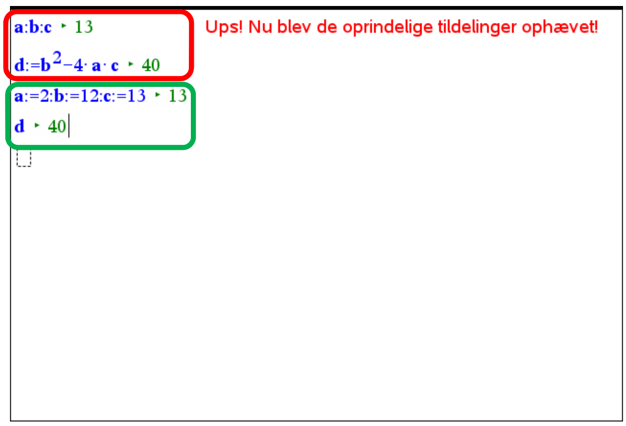
Obs
Gemmer du en ny værdi i en variabel, vil den gamle værdi blive slettet. Indholdet af en variabel kan ses ved blot at skrive variablens navn i et matematikfelt efterfulgt af **Enter**.

Obs
I matematikfelter bliver variable, der har fået tildelt værdier, noteret i **fed**, symbolske variable noteret i *kursiv* og matematikkommandoer i normal skrift.

Det er vigtigt at skrive gangetegn mellem *a* og *c* i udtrykket $b^2 - 4a \cdot c$. Udelader du gangetegnet, opfattes *ac* som navnet på en variabel. Derimod behøver du ikke at skrive noget gangetegn mellem 4 og *a* — her benyttes underforstået multiplikation. Kravene til et variabelnavn gør dette muligt, idet et variabelnavn skal starte med et bogstav.

TI-Nspire CAS genkender godt nok underforstået multiplikation, men det kræver så et *mellemrum* mellem de to variabelnavne. Det ville altså have været korrekt at skrive $d := b^2 - 4a\ \ c$. Så snart du trykker **Enter** indsætter TI-Nspire CAS selv det underforståede gangetegn og alt er som før. Vi vil dog generelt anbefale at man så vidt muligt undlader underforståede gangetegn, da de nemt kan bidrage til forvirring omkring formlernes struktur.

Du kan nu se, at selv om variablen *d* er gemt som en formel, udregnes værdien af *d* som tallet 8. I samme øjeblik du ændrer på værdierne af *a*, *b* og *c*, ændres selvfølgelig også værdien af den afhængige variabel *d*. Ændrer du fx de værdier, der er gemt i variablene *a*, *b* og *c* til *a* = 2, *b* = 12 og *c* = 13, og tjekker *d*'s værdi, ser du, at denne ændres som forventet (grøn indramning):



Men samtidigt ser du også at den oprindelige tildeling af værdier til variablene *a*, *b* og *c* ophæves og de udregnes nu blot med deres aktuelle værdier (rød indramning). Det er i overensstemmelse med det generelle princip:

Det er altid kun den sidst udførte tildeling af en værdi til en variabel, der er gyldig!

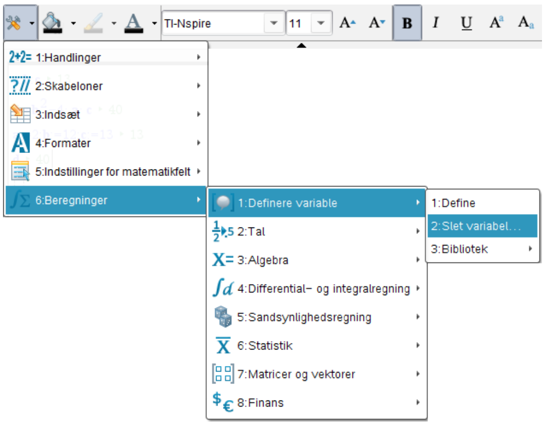
Det er derfor man skal være forsigtig med navngivningen af variable. Hvis man ved et uheld indfører en variabel med samme navn som på en tidligere side *indenfor den samme opgave* ophæves den oprindelige definition og den oprindelige side genberegnes i overensstemmelse med den nye definition: *The past is not protected!*

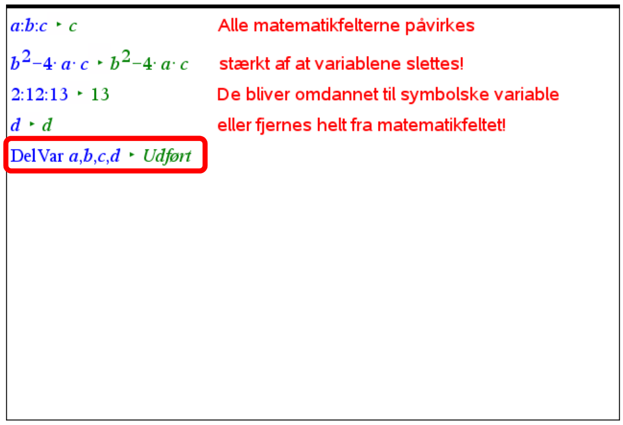
2. Sletning af variable

Du skal nu lave det hele en gang til, men i omvendt orden. Først skal du slette de værdier, *a*, *b*, *c* og *d* har fået tildelt. Du kan nemt klare sagen ved at vælge

∫Σ 6:Beregninger ▸ 1:Definere variable ▸ 2:Slet variabel…

— eller blot skrive kommandoen **DelVar** direkte (rød indramning):



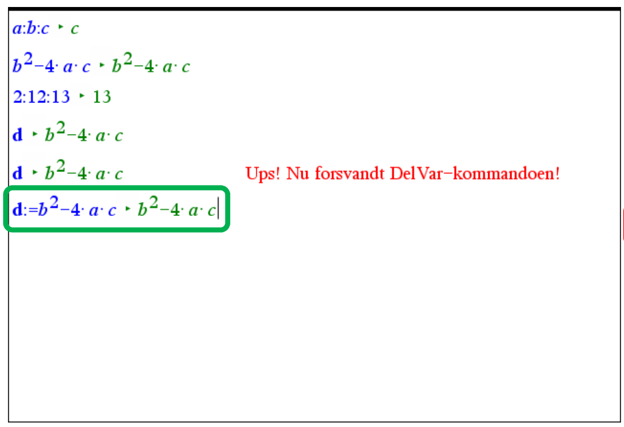


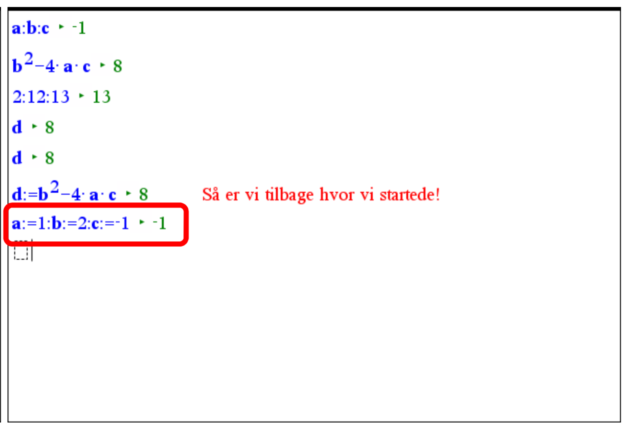
Obs
Husk at en variabel kan kun slettes med DelVar-kommandoen. Den forsvinder ikke bare fordi man sletter et matematikfelt!

Virkningen er som det ses dramatisk: De uafhængige variable *a*, *b* og *c* genopstår som rent symbolske variable uden en værdi. Den afhængige variabel *d* genopstår også som en symbolsk variabel uden værdi, idet sammenhængen til variablen *a*, *b* og *c* nu er brudt. De forskellige tildelinger af værdier degenerer til deres talværdier: De har ikke længere gyldighed som værdier for variablene, da disse nu er slettet fra listen over variable i den pågældende opgave.

3. Tildeling af formler til variable

Nu er variablene *a*, *b* og *c* udefinerede symbolske variable, og når du igen udfører tildelingen $d := b^2 - 4a \cdot c$ (grøn ramme) til den afhængige variabel *d* er det derfor udtrykket $b^2 - 4a \cdot c$, der gemmes i *d*, mens variablen *d* ikke i sig selv har nogen værdi, før de uafhængige symbolske variable *a*, *b* og *c* tildeles værdier (rød ramme).





En ulempe er naturligvis, at så snart du tildeler værdier til a, b og c, vil du ikke længere umiddelbart kunne se, hvilken formel der er gemt i d — det kan du kun, hvis a, b og c er udefinerede.

4. Midlertidig tildeling af værdier til variable

Slet nu variablene a, b og c med kommandoen *DelVar a, b, c* – se venstre skærmbillede nedenfor (grøn ramme). Udregningerne af d opdateres nu, idet maskinen svarer ved at give dig formlen, der er gemt i d (se udregningerne oven over **DelVar** kommandoen).

Du kan nu udføre en *midlertidig tildeling* af værdier til variablene i d og beregne værdien af d ved at skrive:

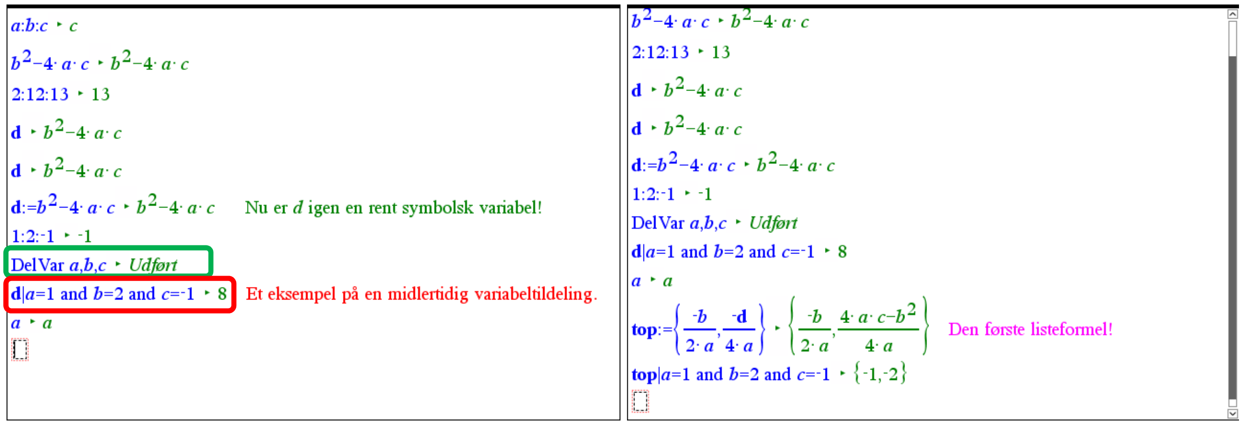
$$d \mid a = 1 \text{ and } b = 2 \text{ and } c = -1$$

I denne kommando står den lodrette streg | for **with**-kommandoen, der kan læses som 'givet at' eller 'hvorom det gælder at', sådan som du måske kender det fra mængdenotationen. Hele kommandoen kan altså læses således

Udregn værdien af d givet at a har værdien 1, b har værdien 2 og c har værdien -1.

Under selve udregningen vil de symbolske variable a, b og c da midlertidigt tildeles de ovenfor nævnte værdier (se igen venstre skærmbillede nedenfor i rød ramme).

Efter denne midlertidige tildeling kan du let tjekke, at du stadig kan fremkalde formlen i d, samt at a, b og c stadigvæk er udefinerede symbolske variable (se igen venstre skærmbillede nedenfor i rød ramme, hvor fx værdien af a tjekkes i den efterfølgende linje).



For at lave en formel, der bestemmer toppunktets koordinater, behøver du nu blot at taste følgende:

$$top := \left\{ -\frac{b}{2a}, -\frac{d}{4a} \right\} .$$

De krøllede parenteser { og }, er vigtige at få med, men betydningen skal du ikke bekymre dig om lige nu. Her bemærker vi blot at variablen **top** er en afhængig variabel, idet dens værdi afhænger af variablene a, b, c og d, samt at værdien af **top** er et koordinatsæt, dvs. en kombination af to tal,

$$\text{nemlig } x\text{-koordinaten} \quad x = -\frac{b}{2a} \quad \text{og} \quad y\text{-koordinaten} \quad y = -\frac{d}{4a} .$$

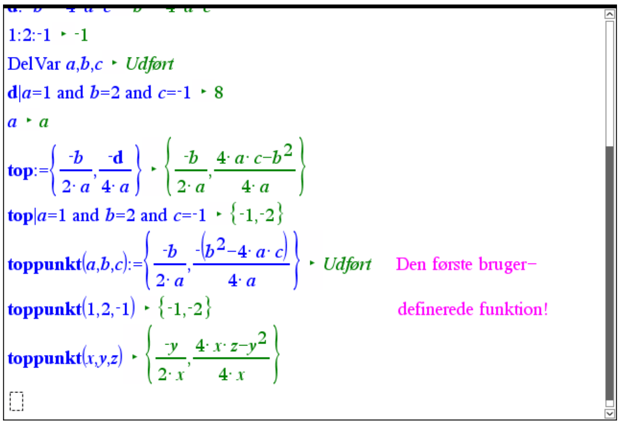
Herefter skal du blot lave en midlertidig tildeling af værdier til variablene i formlen *top* for som vist at få beregnet toppunktets koordinater i et konkret eksempel.

5. En brugerdefineret funktion

Toppunktsformlen kan også implementeres som en brugerdefineret funktion af 3 variable. I **Note**-værkstedet skriver du da (se skærmbillede grøn ramme):

$$toppunkt(a,b,c) := \left\{-\frac{b}{2a}, -\frac{b^2-4a\cdot c}{4a}\right\}$$

Herved defineres en funktion med navnet *toppunkt*. For at bestemme toppunktet for parablen $y = x^2 + 2x - 1$, indtaster du **toppunkt**(1,2,-1):



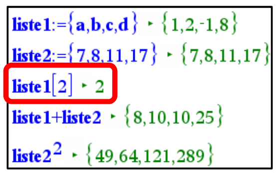
Læg mærke til at navnene på inputvariablene for en funktion kun er *midlertidige* variabelnavne, der oprettes når du skal udregne en funktionsværdi, og som *midlertidigt* tildeles værdierne, som står på deres plads, når du beder om at få en funktionsværdi udregnet, fx **toppunkt**(1,2,–1). Man kan derfor bruge hvilke som helst navne for disse inputvariable, men man bruger normalt *neutrale navne*, altså ikke navnene for allerede navgivne variable med allerede tildelte værdier, selv om det i de fleste tilfælde ikke ville spille nogen rolle.

6. Lister

I ovenstående indskrivning af toppunktsformlen satte du krøllede parenteser om toppunktets koordinater — du lavede en *liste*. Vi vil nu se nærmere på lister. De følgende indtastninger ligger altså i forlængelse af de foregående! En liste er en samling af objekter (tal, udtryk/formler, strenge/tekster), som ikke nødvendigvis er relaterede. Som mængder angives lister med krøllede parenteser og de enkelte elementer separeres af et komma, men til forskel fra mængder, kan en liste indeholde dubletter.

Lister kan indtastes manuelt eller være resultat af anvendelse af en operation, der returnerer en liste - fx *zeros*, som du har stiftet bekendtskab med tidligere i eksempelsamlingens første del. Nedenfor er vist nogle eksempler på lister:

Tip
Læg mærke til, at formlen for *d* udnyttes ved udregning af **liste1**.



På skærmbilledet er vist, hvordan du kan trække et specifikt element, her det andet element ud af **liste2**, ved at sætte en kantet parentes om elementets nummer (rød indramning). Du kan regne på lister præcis som på tal og herunder benytte alle standardfunktioner, selvom det naturligvis stiller visse krav til listens elementer.

1: Variabelstyring

Oversigt over variable

Hvis du har arbejdet de foregående afsnit igennem, så har du haft ganske mange variable i sving: Simple talvariable (uafhængige variable), udtryk/formler (afhængige variable), funktioner og lister. Vi vil nu kigge lidt nærmere på hvordan TI-Nspire CAS håndterer variable for at få en bedre fornemmelse for hvordan de kan udnyttes. Vi vender senere tilbage til toppunktsformlen for en parabel som det gennemgående eksempel, hvor vi vil gå dybere ind i hvordan formlen kan sættes i sving i de forskellige værksteder. Men først vil vi prøve, at danne os et bedre indblik i hvordan variable håndteres i TI-Nspire CAS.

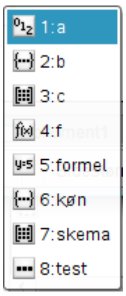
Når man tildeler en værdi til en variabel ved hjælp af tildelingskommandoen := gemmes variablen i et *variabelregister* (også kaldet et symbolregister), der hører til den pågældende opgave. Separate opgaver opretter altså separate variabelregistre. Men forskellige sider indenfor den samme opgave trækker på det samme variabelregister.

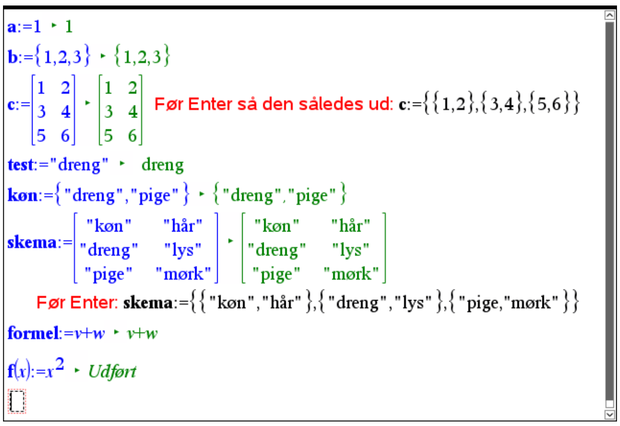
Når TI-Nspire CAS forsøger at reducere/udregne et matematisk udtryk skelner programmet mellem tre forskellige komponenter: *Symbolske variable* (variable, der ikke er tildelt en værdi), **konkrete variable** (variable, der er tildelt en værdi) samt MATEMATIKKOMMANDOER. *Symbolske variable* forbliver uberørte, **konkrete variable** erstattes af deres værdier og MATEMATIKKOMMANDOER udføres efter de gældende algoritmer for den pågældende matematikkommando. For at kunne anvende variablene tilknyttet en opgave er det derfor vigtigt at have overblik over hvilke konkrete variable, der pt ligger på køl i variabelregistret ☺.

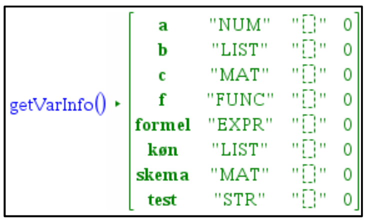
Du kan få en oversigt over de variable, du har i sving, ved at taste var, ved at bruge genvejstasten **Ctrl/Cmd L** (for Listen over variablene) eller ved at bruge kommandoen **GetVarInfo()**.

De to første metoder giver blot listen over **variabelnavnene**, mens den sidste giver en oversigt over variablenes **attributter**, dvs. deres navn, type, bibliotekstilknytning og låsestatus. Variable har nemlig attributter på samme måde som geometriske figurer har det. Men man kan ikke få adgang til deres attributter ved at højreklikke på en variabel: Man er nødt til at spørge generelt via **GetVarInfo()**-kommandoen (eller bruge specialkommandoer til at trække type, biblioteksadgang og låsestatus ud for de enkelte variable en ad gangen, men det vil vi ikke komme nærmere ind på her).

For at frembringe den næste skærmside skal du begynde på en ny opgave ☺







Af variabellisten fremgår, at der pt. er 8 variable i brug. Af piktogrammerne foran variabelnavnene kan du se, hvilken type variable, der er tale om, dvs. hvilken slags værdier den pågældende variabel antager.

1: Variabelstyring

Tip
Du kan altid benytte var tasten eller genvejstasten **Ctrl/Cmd L** til at indsætte (lange) variabelnavne – det er langt hurtigere (og også langt sikrere!) end at skrive dem selv. Du piler blot ned til (eller klikker på) den variabel, du vil indsætte, og taster **Enter**.

Lad os prøve at se lidt nærmere på hvilke typer variable, der understøttes af TI-Nspire CAS. Først og fremmest skelner vi mellem *numeriske variable* og *kategoriske variable*, dvs. variable, hvis værdier er tal, og variable, hvis værdier er tekststrenge. Variable kan være 0-, 1- eller 2-dimensionale. Det svarer i geometrien til punkter, linjer og flader.

Numeriske variable kan være 0-dimensionale, dvs. indeholde et enkelt tal (fx a:=1), 1-dimensionale, dvs. indeholde en liste af tal (fx b:={1,2,3}, eller 2-dimensionale, dvs. indeholde en tabel med tal (fx c:={{1,2},{3,4},{5,6}}).

Kategoriske variable kan tilsvarende være 0-dimensionale, dvs. indeholde en enkelt tekststreng (fx test:=”dreng”), 1-dimensionale, dvs. indeholde en liste af tekststrenge (fx køn:={”dreng”,”pige”}), eller 2-dimensionale, dvs. indeholde en tabel med tekst-strenge (fx. Skema:={{”køn”,”hår”},{”dreng”,”lys”},{”pige”,”mørk”}}

En *talvariabel* har typen ”NUM” (for Number) med piktogrammet (læg mærke til tallene 0, 1 og 2).

Obs
Ved tekstvariable skal man være omhyggelig med at sætte gåseøjne korrekt: Man må *ikke* sætte gåseøjne inde i gåseøjne: Først afsluttes de gåseøjne man er i gang med, så piler man videre og sætter nye gåseøjne!

En *tekstvariabel* har typen ”STR” (for Streng) med piktogrammet (læg mærke til ellipsen …).

En *listevariabel*, hvad enten den indeholder tal, tekst eller formler har typen ”LIST” med piktogrammet (læg mærke til de krøllede parenteser).

En *tabelvariabel*, hvad enten den indeholder tal, tekst eller formler har typen ”MAT” (tabeller hedder også Matricer!) med piktogrammet (læg mærke til de kantede parenteser).

Tip
Tabeller indsættes nemmest ved hjælp af tabelskabelonen fra værktøjet **matematikskabeloner** i venstre sidepanel.

En *formelvariabel*, der bygger på et udtryk indeholdende symbolske variable, har typen ”EXPR” (for Expression) med piktogrammet (læg mærke til lighedstegnet).

En *funktionsvariabel* har typen ”FUNC” med piktogrammet (læg mærke til funktionsparentesen).

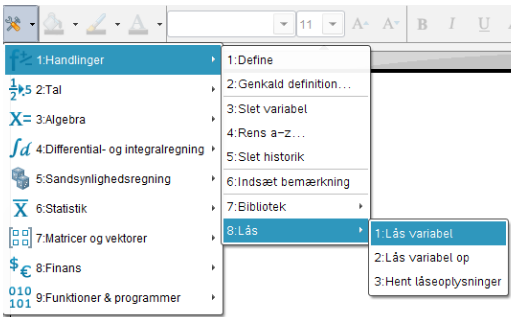
En *programvariabel* har typen ”PRGM” med piktogrammet (læg mærke til de binære koder). Vi vil ikke beskæftige os med programmer i dette hæfte.

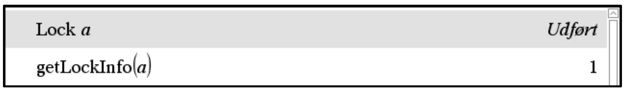
Kigger vi i **GetVarInfo**-listen over attributter, dukker der ud over navn og type to attributter mere op: **Biblioteksadgang** og **Låsestatus**.

Ingen af de variable vi har oprettet har *biblioteksadgang*, men det er altså i princippet muligt at gemme variable, fx matematiske konstanter som det gyldne snit, i et biblioteksdokument, og så trække dem ind i andre dokumenter/opgaver. Biblioteksadgang er ikke noget vi vil komme yderligere ind på i dette hæfte.

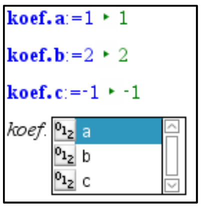
Til gengæld kan *låsestatus* være vigtig. I almindelighed er data *ikke* beskyttede. Hvis vi indtaster variable, er der mange måder de kan ændres på, fx ved at man trækker i datapunkter i et tilhørende diagram. Sådanne ændringer kan være utilsigtede og det er for at beskytte mod utilsigtede ændringer at man kan låse en variabel (hvorved låsestatus skifter fra 0 til 1). Dette kan *kun* gøres i et **Beregnings**-værksted og det sker med **Lock**-kommandoen:

1: Variabelstyring





Navngivning af variable



En variabel kan både have sit *eget navn* og et *familienavn*, men typisk bruger vi kun 'fornavnet'. Hvis der også skal være et familienavn sættes det *forrest* og det adskilles fra fornavnet ved et punktum. Man kan altså fx give koefficienterne til et andengrads-polynomium navnene:

koef.a, *koef.b* og *koef.c*

for at markere at de er i familie med hinanden. Når man skal referere til disse variable kan man da nøjes med at skrive koef., hvorved der når man taster punktummet åbnes for en rulleliste af mulige variabelnavne indenfor familien.

Vi vender om lidt tilbage til variable *med* familienavne, men ser i det følgende på variable *uden* familienavn. Her kan navnet bestå af op til 16 tegn, tal og bogstaver og understregningstegnet _. Et variabelnavn skal være sammenhængende, dvs. det må ikke bestå af adskilte ord. Man kan imidlertid sætte en _ i stedet for mellemrummet.

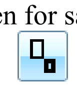
Et variabelnavn kan *ikke* starte med et tal, men ellers er opbygningen meget fleksibel. Der kan anvendes almindelige latinske bogstaver, der kan anvendes særlige nationale bogstaver, så som æ, ø og å, ligesom der kan anvendes bogstaver med accent. Endelig kan der anvendes indices, dvs. dele af variabelnavnet kan skrives med *sænket skrift*. Det er altså helt i orden med navne som

x_{top} og y_{top}

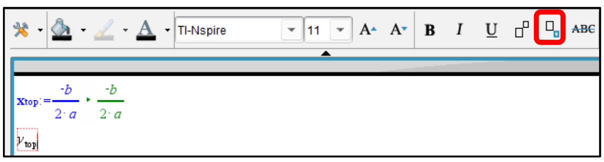
(som er noget helt andet end variablene med navnene *xtop* og *ytop*). Hvis det drejer sig om simple indices fx

x_1 og y_1

så er det nok nemmest at skrive indices ved hjælp af den nederste del af tegn-oversigten. Men ellers må man bruge skabelonen for sænket skrift:



Der findes nemlig ingen tastaturgenvej til sænket skrift. Læg mærke til at skabelonen for sænket skrift også findes på formateringsbjælken foroven ☺.



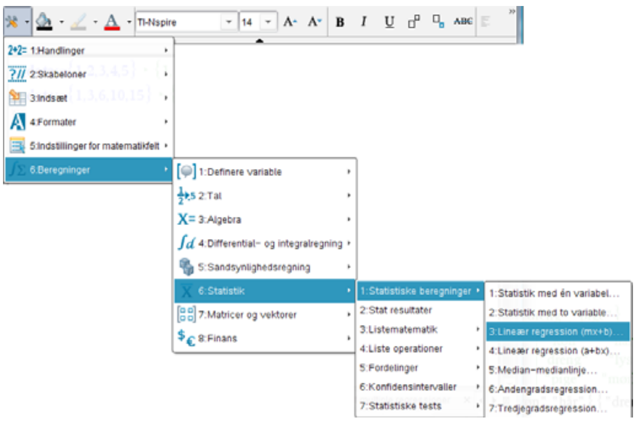
Når man navngiver variable er det vigtigt at man undgår at bruge allerede reserverede navne på fx matematikkommandoer, grafer, koordinater osv. Det er fx *ikke* en god ide at kalde en variabel for sum, da dette allerede er navnet på en matematikkommando.

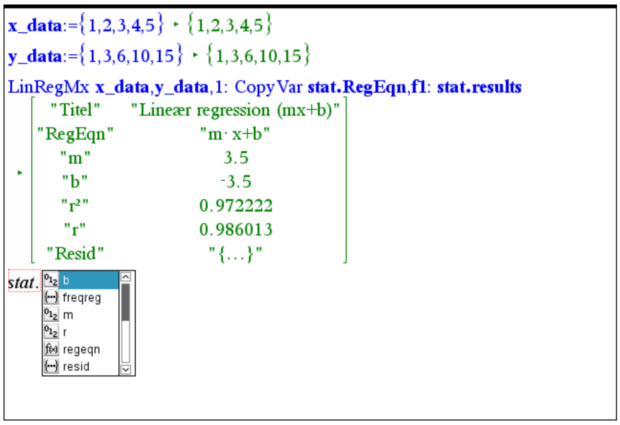
Man kan tjekke om det er navnet på en matematikkommando ved at se om programmet skriver det i NORMAL SKRIFT.

Det er heller ikke en god ide at kalde en variabel for x1 eller y2, da disse navne er reserverede til grafer for parameterfremstillinger. I stedet bør man kalde dem x_1 og y_2 eller benytte understregningstegnet til at skille bogstav og tal, dvs. kalde dem x_1 og y_2.

Endelig må det kraftigt frarådes at bruge aksebetegnelser som *x*, *y* og *z* eller parameterbetegnelser som *t*, θ og *u* som variabelnavne. Det kan have helt utilsigtede konsekvenser, da disse navne også indgår i forskrifter for funktioner og parameterfremstillinger!

Vi vender nu kortvarigt tilbage til variable med familienavne. Sådanne variable oprettes automatisk af TI-Nspire CAS under bestemte omstændigheder. Fx oprettes der helt automatisk en stribe statistiske variable med familienavnet **stat.**, når man udfører en statistisk beregning, fx en lineær regression.





Det er nu afgørende at forstå at disse variable overskrives næste gang man udfører en statistisk beregning indenfor den samme opgave. Hvis man vil beskytte resultatet af den statistiske beregning må man derfor gå omhyggeligt frem! Der er to forskellige strategier: I **Lister og regneark**-værkstedet beskyttes resultatet automatisk ved hjælp af en **CopyVar** kommando, der overfører indholdet af de statiske variable fra den lineære regression til nye statistiske variable:

D =LinRegMx('**x_data**,'**y_data**,1): CopyVar **Stat.RegEqn**,'*f1*: CopyVar **Stat.**, **Stat1.**

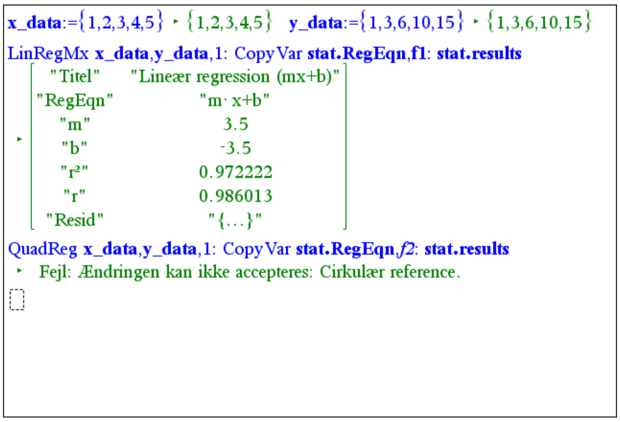
F =QuadReg('**x_data**,'**y_data**,1): CopyVar **Stat.RegEqn**,'*f2*: CopyVar **Stat.**, **Stat2.**

	A x_data	B y_data	C	D	E	F
=				=LinRegMx('x		=QuadReg('
1	1	1	Titel	Lineær regr...	Titel	Andengrad...
2	2	3	RegEqn	m*x+b	RegEqn	a*x^2+b*x...
3	3	6	m	3.5	a	0.5
4	4	10	b	-3.5	b	0.5
5	5	15	r²	0.972222	c	-1.ᴇ-13
6			r	0.986013	R²	1.
7			Resid	{1.,-0.5,-1.,...	Resid	{1.ᴇ-14,-6...
8						
9						
10						

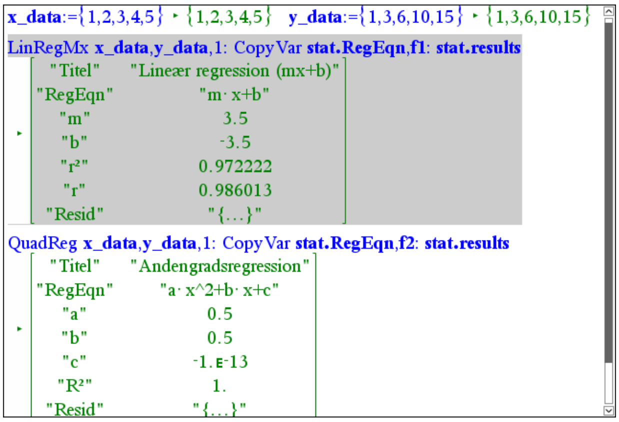
D =LinRegMx('**x_data**,'**y_data**,1) : CopyVar **Stat.RegEqn**,'*f1*: CopyVar *St*

1: Variabelstyring

I **Note**-værkstedet er man ikke så heldig: Her må man selv beskytte den statistiske udregning. Det sker ved at *beskytte* matematikfeltet med resultaterne fra den første statistiske beregning, idet et beskyttet matematikfelt ikke opdateres (med mindre du giver lov til det ☺). Du beskytter et eller flere matematikfelter ved at sværte det/dem til, højreklikke og vælge: **9:Handlinger ▸ 4:Deaktiver**. Uden beskyttelse fås en fejlmeddelelse, når man forsøger at udføre den anden regression.



Uden beskyttelse!



Med beskyttelse!

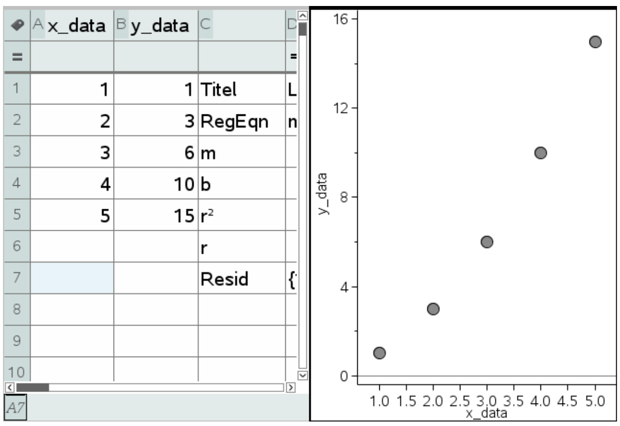
Automatisk oprettelse af konkrete variable

I **Beregnings**-værkstedet og **Note**-værkstedet sker tildeling af værdier til en variabel typisk ved hjælp af en tildelingskommando, typisk :=. I de øvrige værksteder er der derimod forskellige mere skjulte procedurer for at indføre navngivne variable med værdier, som *ikke* kræver brug af tildelingskommandoen := . Det kan være nyttigt at kende de vigtigste af disse tilfælde:

Obs
Husk at variable, der stammer fra navngivne søjler *ikke* forsvinder fra variabelregistret bare fordi man ændrer navnet på søjlen eller evt. sletter søjlen helt. De kan kun fjernes fra variabelregistret ved hjælp af **DelVar**-kommandoen!

1. Lister og regneark

I **lister og regneark**-værkstedet arbejder man typisk med *listevariable*. I samme øjeblik man navngiver en søjle i regnearket, oprettes der en listevariabel med dette navn i variabelregistret. Listen er tom indtil man begynder at indtaste værdier i søjlens celler eller frembringer værdierne via en formel i formelfeltet lige under navnefeltet. *Navngivne søjler* er derfor tilgængelige fra andre værksteder og det er fx fuldstændigt afgørende for at man kan oprette grafer over data i regnearket. Den eneste måde fx **Diagrammer og Statistik**-værkstedet kan håndtere grafer for data er, hvis det kan hente de fornødne data i variabelregistret. I **Lister og regneark**-værkstedet er det tilrådeligt at navngive variable med mere end et bogstav, da det ellers kan forveksles med søjlenavne (eller cellebetegnelser).



Tilsvarende kan man *kæde* indholdet af en eksisterende listevariabel til en søjle ved simpelthen at skrive variablens navn i navnefeltet eller højreklikke i det tomme navnefelt og vælge menupunktet

3: Variable ▸ 3: Kæd til ▸ (liste over tilgængelige listevariable)

Når vi kigger på de enkelte celler så kan de jo indeholde et tal, en tekst (skrevet i gåseøjne), en formel (skrevet med et lighedstegn) eller et matematisk udtryk indeholdende symbolske variable (dvs. tekst og formler uden gåseøjne eller lighedstegn). Det sidste er ikke så velkendt så lad os lige se et enkelt eksempel på anvendelsen af et symbolsk regneark: Hvis *a* og *b* er symbolske variable, dvs. de er ikke tilskrevet nogen værdi og står derfor ikke i variabelregistret, kan vi blot indskrive *a* i celle A1 og *b* i celle A2. I celle A3 kan vi så indskrive formlen = A1+A2, og trække den ned gennem regnearket, dvs. enhver ny celle er netop summen af de to foregående celler. Celleformlerne vil da blive udregnet som symbolske udtryk. Læg mærke til koefficienterne, som ikke overraskende afspejler Fibonaccitallene ☺.

	A	B	C	D	E
=					
1	a				
2	b				
3	a+b				
4	a+2*b				
5	2*a+3*b				
6	3*a+5*b				
7	5*a+8*b				
8	8*a+13*b				
9	13*a+21*b				
10	21*a+34*b				

A3 =a1+a2

Men tilbage til en enkelt celle i en søjle, hvor søjlen *ikke* er navngivet eller drevet af en søjleformel: Indholdet af cellen kan da *lagres som en variabel* ved hjælp af tildelingskommandoen :=, dvs. ved simpelthen at skrive følgende i cellen

variabelnavn:=udtryk,

Eller vi kan højreklikke i cellen og vælge menupunktet

7: Variable ▸ 1: Gem var.

	A x_data	B y_data	C	D	E	F
=				=LinRegMx('x		
1	1	1	Titel	Lineær regr…		Titel
2	2	3	RegEqn	m*x+b		RegEc
3	3	6	m	3.5	**3.5**	a
4	4	10	b	-3.5		b
5	5	15	r²	0.972222		c
6			r	0.986013		R²
7			Resid	{1.,-0.5,-1.,…		Resid
8						
9						
10						

E3 **slope**:=d3

Her har vi således gemt hældningen for den rette linje (celle d3) i variablen **slope**.

På samme måde kan vi *kæde* indholdet af en tom celle til en eksisterende simpel variabel ved at skrive følgende i cellen

= variabelnavn

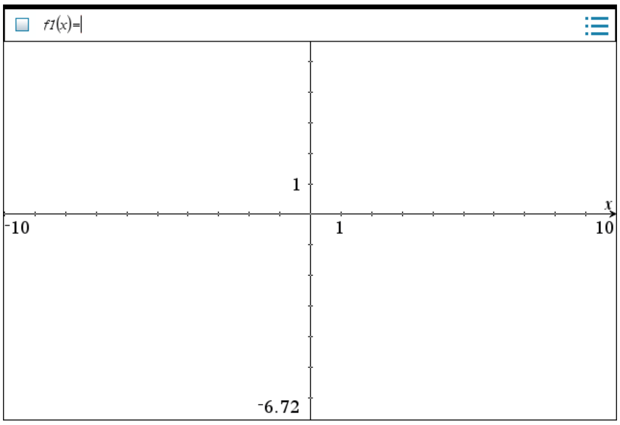
Eller vi kan højreklikke i cellen og vælge menupunktet

7: Variable ▸ 3: Kæd til ▸ (liste over tilgængelige simple variable)

1: Variabelstyring

Der er altså rige muligheder for at lagre resultater fra regnearket i variable, der kan bruges i andre værksteder, eller lade regnearket afspejle variable i andre værksteder. **Lister og regneark** er således i høj grad et fuldbårent symbolsk interaktivt dynamisk regneværksted med kvaliteter, der fuldt ud matcher de to andre regne-værksteder: **Beregnings**-værkstedet og **Note**-værkstedet.

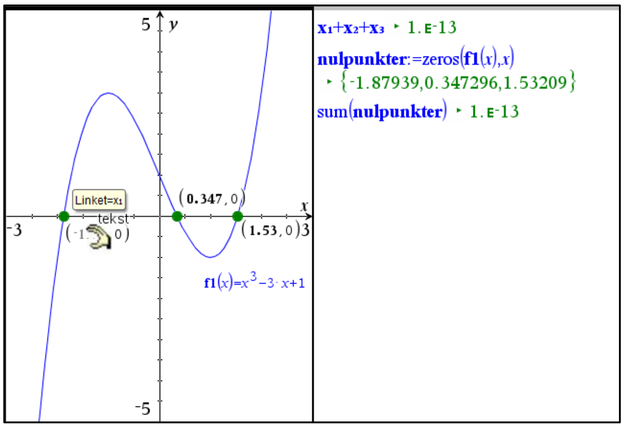
2. Grafer



Obs
Hvis man fjerner en forskrift fra en funktion på graflisten forsvinder funktionen *også* fra variabelregistret.
For én enkelt gangs skyld er det altså *ikke* nødvendigt at bruge **DelVar** for at fjerne en variabel fra variabelregistret ☺

Når vi opretter en graf via indtastningslinjen lagres forskriften automatisk som en funktionsvariabel med det navn, den er blevet tildelt, dvs. typisk f1(x), f2(x), ...(men du kan også selv vælge navnene for dine funktioner). Det samme gælder for de øvrige graftyper (fx punktplot og parameterfremstillinger). Forskrifterne for alle de funktioner, vi tegner i grafrummet havner altså automatisk i variabelregistret og er derfor automatisk til rådighed for andre værksteder, ikke mindst **Note**-værkstedet.

Derudover kan man lagre fx målinger og koordinater i navngivne variable. Det sker ved at højreklikke i målingen/koordinaten og vælge menupunktet **lagre**. Hvis vi fx har fundet nulpunkterne for et tredjegradspolynomium kan vi efterfølgende lagre dem i variablene x_1, x_2 og x_3 og efterfølgende beregne deres sum:

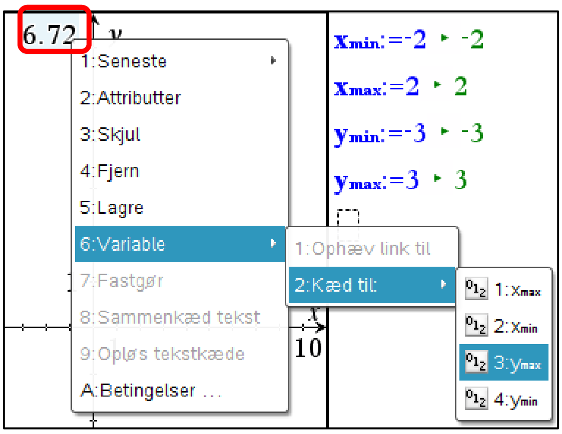


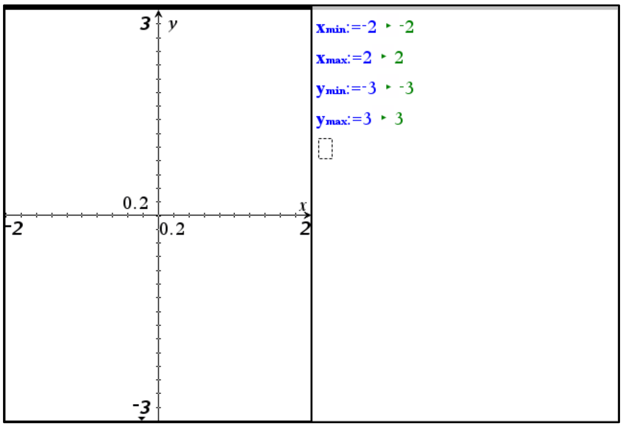
Tilsvarende kan man kæde fx målinger og koordinater til navngivne variable. Det sker ved at højreklikke og vælge menupunktet:

Variable ▸ Kæd til ▸ (liste over tilgængelige simple numeriske variable)

Det sidste åbner for eksempel muligheden for som vist at kæde aksernes start- og slutværdier til navngivne variable og dermed til at styre udstrækningen af koordinatsystemet fra andre værksteder ☺

1: Variabelstyring



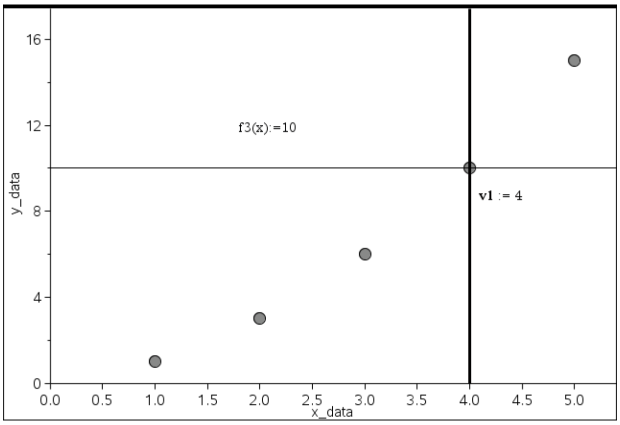


Her kædes til y_{max}.

Her er akseværdierne kædet til x_{min}, x_{max}, y_{min} og y_{max}!

3. Diagrammer og statistik

Endelig er der tegneværkstedet **Diagrammer og statistik**, hvor man afbilder diverse diagrammer for allerede eksisterende liste-variable.



Men man kan også tilføje grafer for funktioner eller plotte værdier som lodrette linjer ved hjælp af menupunktet **4:Undersøg data**. Forskrifterne for disse funktioner lagres da automatisk som funktionsvariable, ligesom de plottede værdier lagres automatisk som simple numeriske variable.

*Gensyn med parablen**

Vi vender nu som lovet tilbage til toppunktsformlen for en parabel

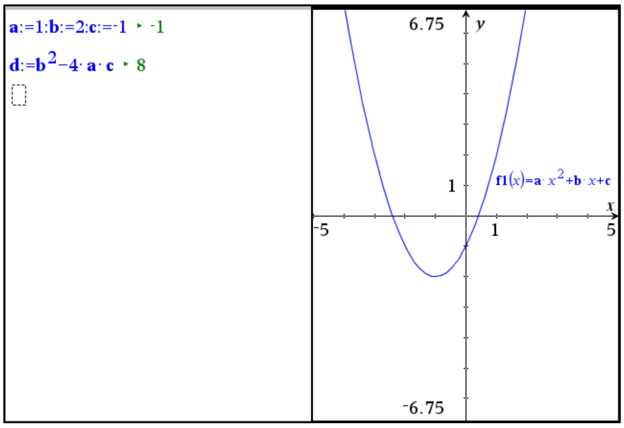
$$y = a \cdot x^2 + b \cdot x + c$$

med diskriminanten

$$d = b^2 - 4a \cdot c$$

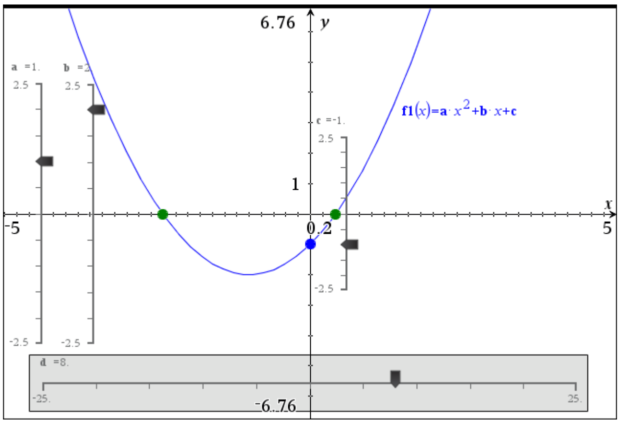
Denne gang inddrager vi også et **Graf**-værksted, så man også kan se parablen og ikke bare dens ligning ☺. Vi arbejder os nu hen mod diverse interaktive illustrationer af dels koefficienterne *a*, *b* og *c*'s betydning, dels betydningen af diskriminanten *d*.

1: Variabelstyring



Det første trick er at håndtere variablene som skydervariable, så vi kan trække i dem. Skyderen er den arketypiske repræsentation af en numerisk variabel, hvor man for alvor kan se og føle, at den er variabel. For bedre at kunne se skyderne splitter vi de to værkstedsruder ad, så **Graf**-værkstedet får sit eget vindue (fx med genvejen **Ctrl/Cmd 6**):

Obs
Hvis du flytter et **Note**-værksted over på en ny side, hvad enten det sker ved en opsplitning af en eksisterende side eller ved at kopiere og indsætte **Note**-værkstedet på en ny side, er **note**-værkstedet som udgangspunkt inaktivt, dvs. matematikfelterne passive. Vælg dem alle (**Ctrl/Cmd A**), højreklik og vælg **9:handlinger ▸ 1: Evaluer det markerede**



Husk at genaktivere formlerne på **Note**-siden, når du splitter værkstederne ad.

Læg mærke til forskellige raffinementer: Skyderne for de uafhængige variable *a*, *b* og *c*, dvs. koefficienterne til andengradspolynomiet er alle afsat lodret. De er også alle afsat, så nulpunktet på skyderens skala netop falder på *x*-aksen. Det er altså visuelt meget nemt at se deres fortegn! Endelig er skyderen for *c* anbragt opad *y*-aksen og skalaen er netop tilpasset *y*-aksens skala, så det visuelt bliver meget tydeligt at *c* netop regulerer skæringen med *y*-aksen. For yderligere at fremhæve denne pointe har vi tilføjet skæringspunktet med *y*-aksen og fremhævet skæringspunktet ved at fede punktet op og farve det blåt.

Trækker vi i skyderne for *a*, *b* og *c* overtager vi nu tildelingen af værdierne til *a*, *b* og *c* fra de tilsvarende tildelinger på **Note**-siden. På denne måde kan vi altid konvertere en almindelig numerisk variabel til en skyder variabel. Vi kan endda oprette flere eksemplarer af den samme skyder som kan operere fra forskellige sider i dokumentet. Hver gang vi trækker i en skyder et eller andet sted i dokumentet er det så denne skyder der overtager tildelingen af værdier til variablen i variabelregistret og alle de andre skydere retter bare ind ☺ .

Læg også mærke til, at vi har afgrænset skyderen for den afhængige variabel *d* af et gråt rektangel. Det skal tjene som en reminder om at vi **ikke under nogen omstændigheder må trække i skyderen for *d***! Gør vi det overtager vi jo kontrollen med tildelingen af værdier til variablen *d*, og så er den ikke længere afhængig! Den skal bare følge med når vi trækker i de andre skydere. Ydermere har vi placeret skalaen for skyderen *d*, så nulpunktet falder på *y*-aksen. Det er altså meget nemt at se om den er negativ eller positiv. Det har som bekendt konsekvenser for skæringen mellem *x*-aksen og parablen: Hvis *d* er negativ er der ingen skæringspunkter. Hvis *d* er nul er der netop et røringspunkt og hvis *d* er positiv er der to skæringspunkter. Vi har derfor tilføjet skæringspunkterne mellem *x*-aksen og parablen og fremhævet dem ved at fede punkterne op og farvet dem grønne.

Prøv nu selv at trække i skyderne for *a*, *b* og *c* og se hvilken indflydelse det har på parablens form og placering.

Den ovenstående illustration er særlig rettet mod at vise betydningen af koefficienten *c*. Det er mere udfordrende at vise betydningen af koefficienterne *a* og *b*!

Her står *a* for parablens krumning eller hulning. Det er nemt at se *a*'s fortegn, der jo afspejler om parablen er glad eller sur. Men det er sværere at se størrelsen af *a*! Det kan dog gøres på forskellige måder. Fx kan man overveje, hvad der sker, når man bevæger sig stykket 1 væk fra toppunktet.

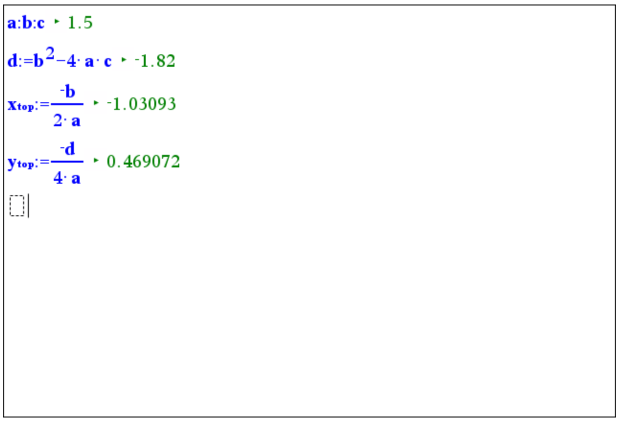
Tilsvarende står *b* for hældningen af grafen i skæringspunktet med *y*-aksen. Man kan da med fordel tilføje tangenten i skæringspunktet, dvs. den lineære del af andengradspolynomiet

$$f2(x) = b \cdot x + c$$

Det gør det muligt at illustrere betydningen af *b* som en hældning.

Vi går ikke ind i detaljer men lader det stå som en udfordring at frembringe passende interaktive illustrationer, der viser betydningen af koefficienterne *a* og *b* ☺

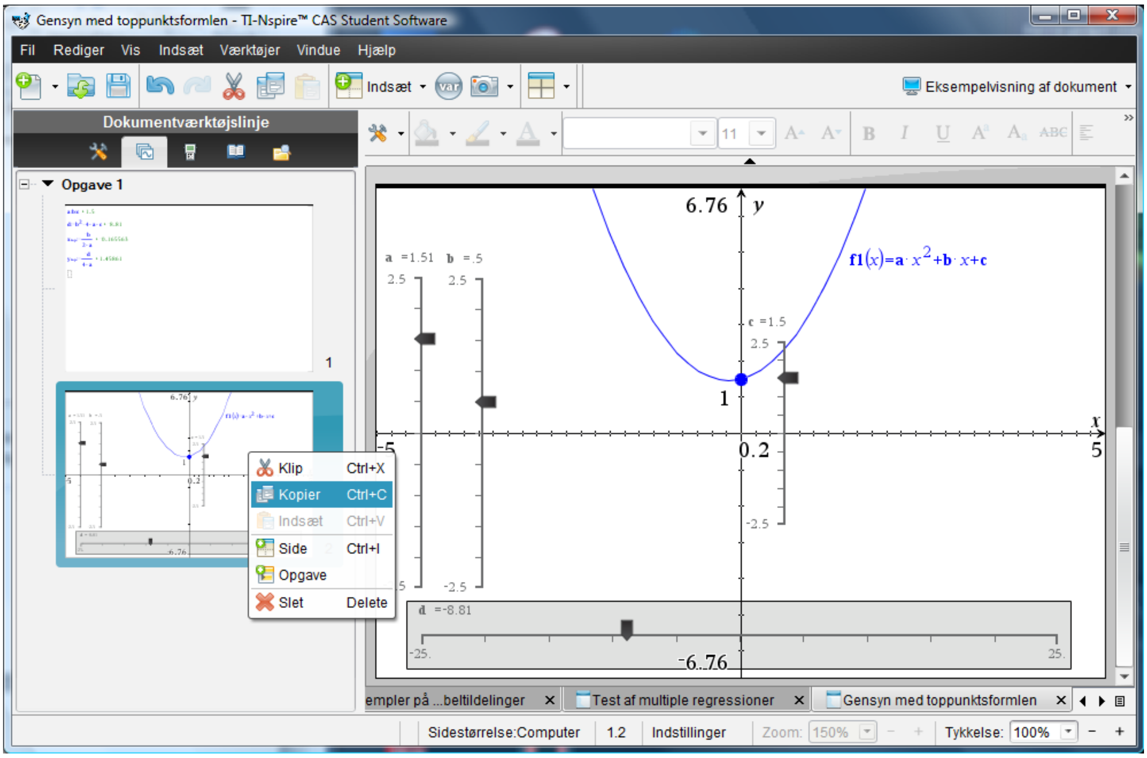
I stedet vender vi os mod toppunktet for parablen: Vi vil altså nu inddrage toppunktsformlerne og tilføjer derfor disse til **Note**-værkstedet.

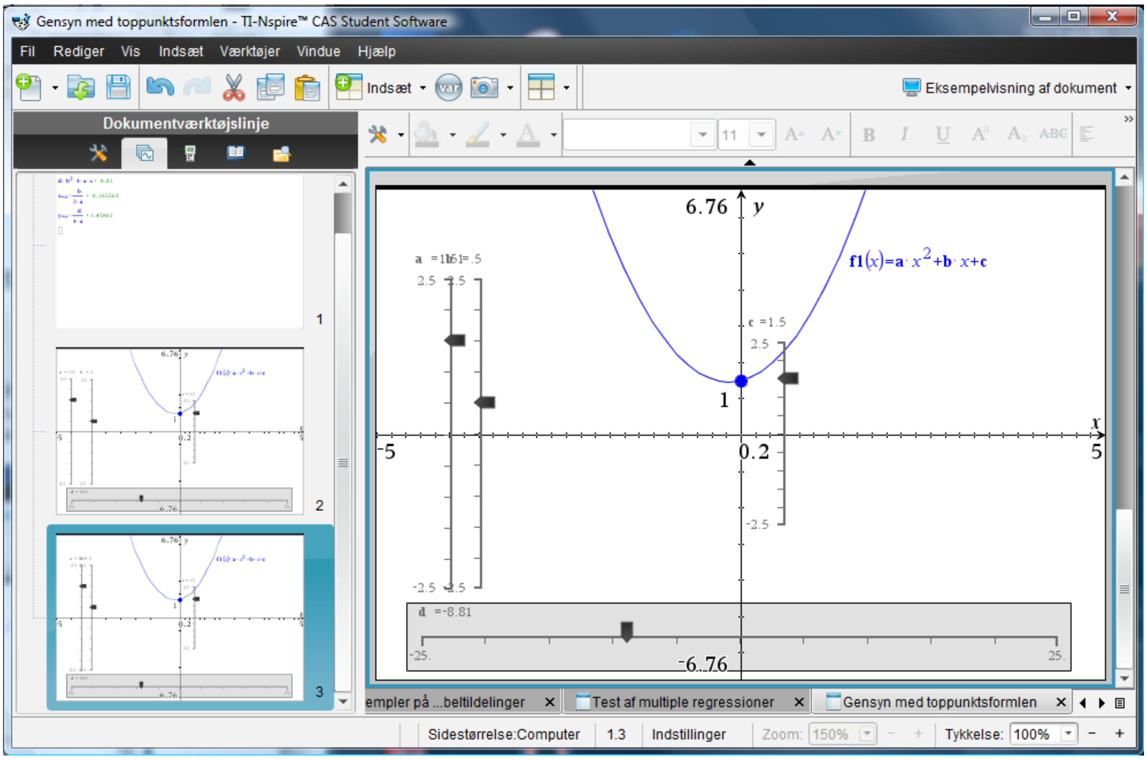


Læg mærke til den første linje: Det er ikke længere **Note**-værkstedet, der styrer værdien af *a*, *b* og *c* ☺. Det første matematikfelt afspejler nu blot den aktuelle indstilling af skyderne for *a*, *b* og *c* (og da det er en multipel kommando ser vi kun den sidste værdi).

Vi skal nu have overført toppunktsformlen til **Graf**-værkstedet. For ikke at ødelægge den **Graf**-side vi allerede har bygget op til illustration af koefficienten *c*, overfører vi denne side til en ny **Graf**-side. Det gøres nemmest ved ude i **sidesortereren** at tage en kopi af graf-siden og så indsætte den som den næste side!

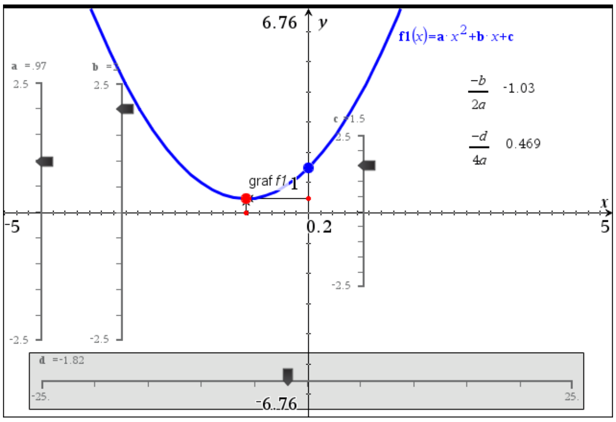
1: Variabelstyring





Vi skal så have overført toppunktsformlerne til graf-værkstedet. Det kan gøres på rigtigt mange måder. Her *kæder* vi simpelthen et frit punkts koordinater til variablene x_{top} og y_{top}. Du indfører altså et frit punkt (så det må fx ikke ligge på parablen, men skal indsættes i et tomt rum på **Graf**-siden), bestemmer dets koordinater og kæder! På figuren har vi fedet toppunktet op og farvet det rødt.

1: Variabelstyring



Vi har ydermere tilføjet tekstbokse med formlerne for *x*-koordianten og *y*-koordinaten og beregnet disse (hvor vi har tastet L for at få indsat værdierne for de variable *a*, *b*, *c* og *d*). Disse værdier kan så overføres til akserne (små røde punkter), og endelig kan vi tilføje lodrette og vandrette vektorer, der peger på toppuntket.

Som altid i TI-Nspire CAS er der mange forskelige måder at opbygge en sådan illustration. Vi har bare antydet nogle få – kun fantasien sætter en grænse for hvordan man kan udnytte interaktiviteten og dynamikken indbygget i TI-Nspire CAS ☺

2 Ligninger og uligheder

Løsning af ligninger og uligheder med Solve-kommandoen

Du har allerede set nogle eksempler på såvel symbolsk som numerisk ligningsløsning med **Solve**-kommandoen, men der er meget mere at se på i den forbindelse:

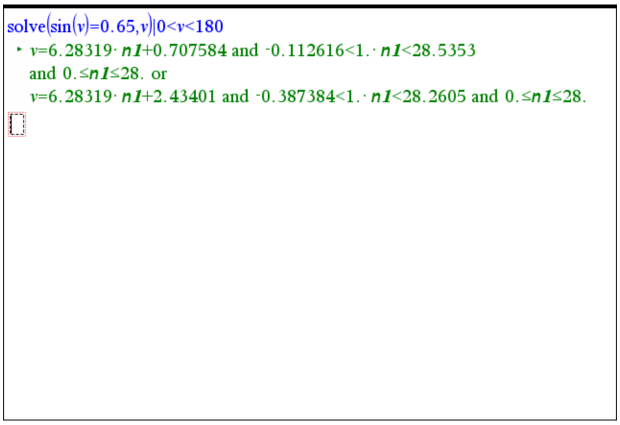
- trigonometriske ligninger
- to ligninger med to ubekendte
- ligninger med parametre
- numerisk nulpunktsbestemmelse
- uligheder

1. Trigonometriske ligninger

Løs ligningen $\sin(v) = 0.65$, hvor $0° < v < 180°$

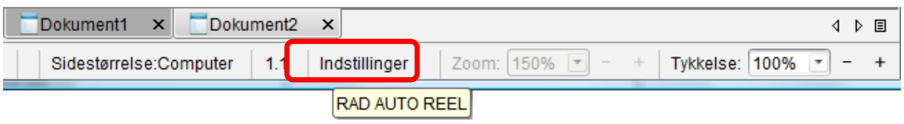
Indsæt et **Note**-værksted og løs ligningen med betingelsen 0<*v*<180:

Tip
Det er ikke sikkert, at du får ***n1*** i løsningen. Næste gang, du løser en ligning af denne type, udtrykkes løsningen ved ***n2*** osv. indtil ***n255***, hvor der startes forfra.



Hvis du får et resultat, som det du ser i skærmbilledet ovenfor, er det fordi din TI-Nspire CAS som generel fabriksindstilling er indstillet til at regne i radianer.

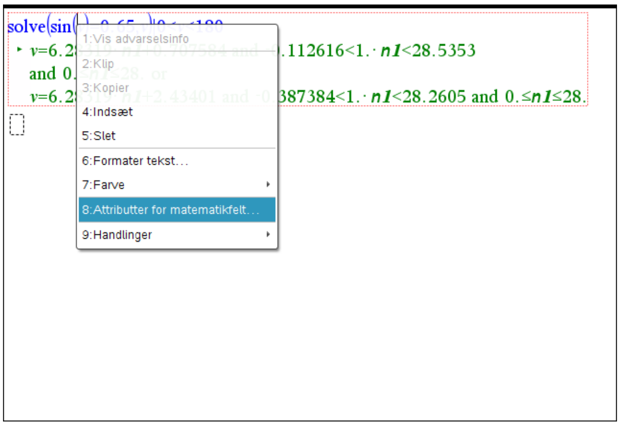
Du kan let tjekke din indstilling ved holde markøren over feltet **Indstillinger** :

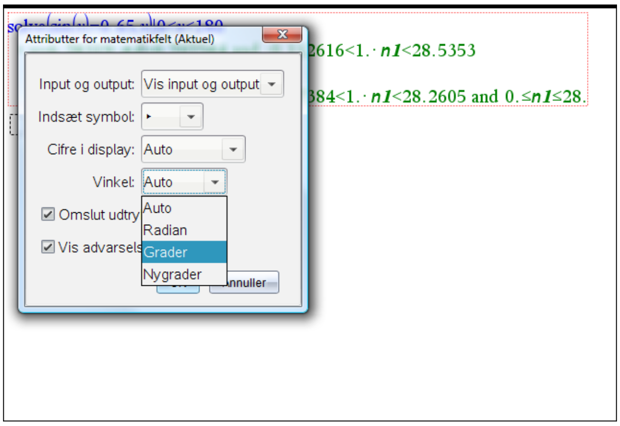


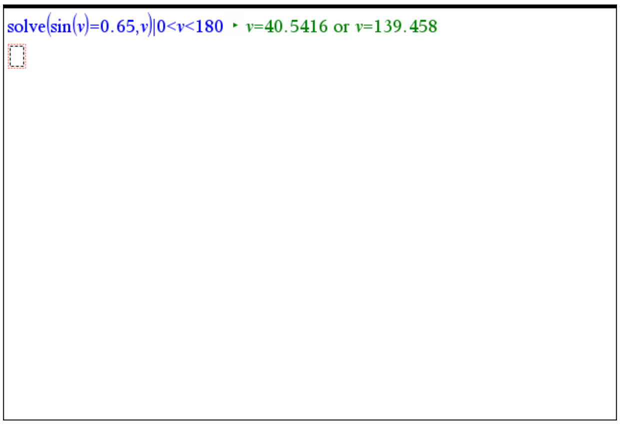
Tip
Du kan også tvinge **Solve**-kommandoen til at finde løsninger i gradtal ved at tilføje et gradtegn til variablen, dvs.
Solve(sin(v°) = 0.65, v)|0<v<180

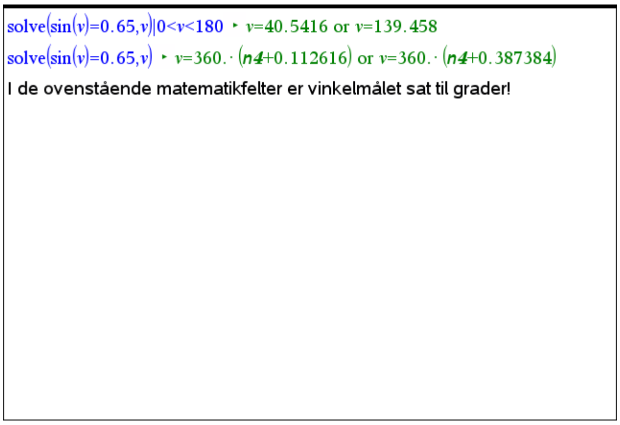
Det er meget nemt at få ændret resultatet til grader i **Note**-værkstedet:

Højreklik på resultatet og vælg **8:Atibutter for matematikfelt**, og skift til grader i den dialogboks der popper op — og du får det ønskede resultat (venstre skærmbillede forneden). Uden betingelsen 0<*v*<180 ser løsningen derimod lidt underligt ud, men viser, hvordan TI-Nspire CAS tackler en ligning med uendelig mange løsninger (højre skærmbillede forneden).





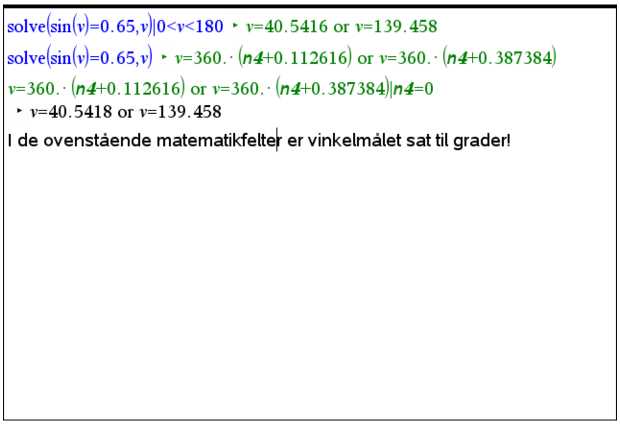




Variablen ***n*4** står for en heltallig konstant.

Løsningerne i intervallet 0<*v*<180 finder du så ved at sætte ***n*4** = 0. Det klarer du sådan: Kopier løsningerne til et nyt matematikfelt og tilføj |***n*4** = 0, idet du også kopierer og indsætter ***n*4**, som *ikke* er en almindelig bogstavvariabel.

Tip
Du kan også hente ***n*** fra symbolpaletten — eller skriv **@*n*4**, hvor **@** fortæller, at der kommer et specialtegn.



Bemærkning: Du kan i princippet også ændre indstillingerne permanent ved at dobbelt-klikke på **Indstillinger** i statuslinjen og ændre **Vinkel** til grader. Men vi vil *ikke* anbefale at man ændrer på de generelle indstillinger, da man så kan have svært ved at sammenligne ens egne resultater med de resultater, de andre i klassen får ☺. Hvis alle kører med de samme fabriksindstillinger, så er der allerede et problem mindre at slås med, når der sker noget uventet.

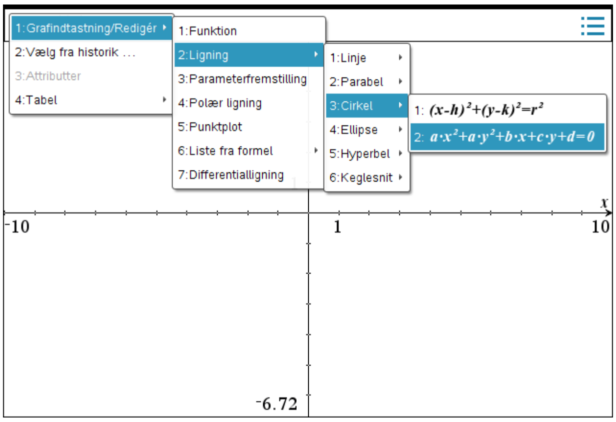
Husk også at du kan tvinge TI-Nspire CAS til at regne i grader ved at tilføje velvalgte gradtegn, fx ved at skrive sin(30°) i stedet for sin(30), hvor du overlader det til TI-Nspire CAS selv at vælge vinkelmålet ☺

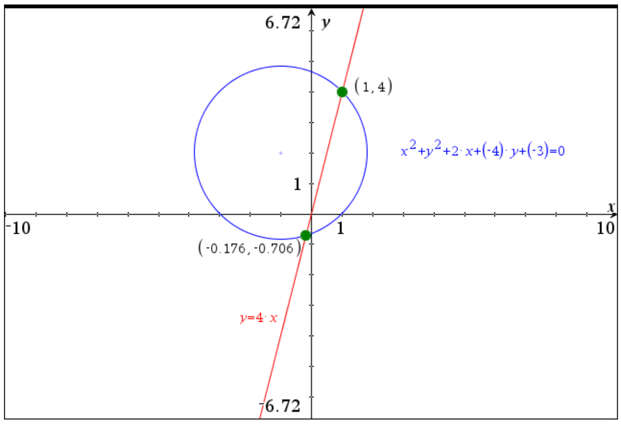
2. To ligninger med to ubekendte

Løs ligningssystemet

$$x^2 + 2x + y^2 - 4y - 3 = 0 \quad \wedge \quad y = 4x$$

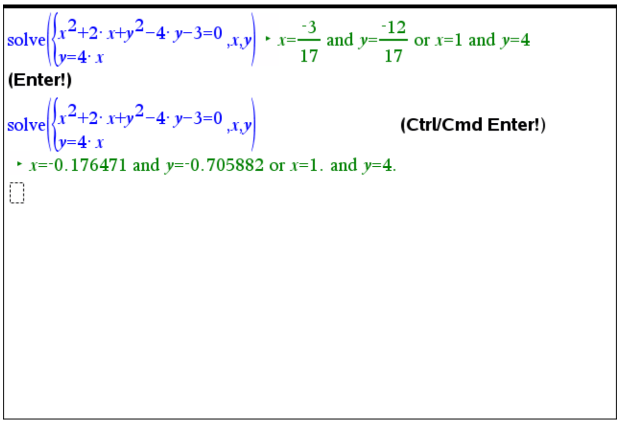
Geometrisk svarer opgaven til at finde skæringspunkterne mellem en cirkel og en linje. Disse kan indtegnes ved brug af graftypen **Ligninger** i et **Graf**-værksted:





Du indtaster nemmest ligningssystemet ved at benytte skabelonen for to ligninger med to ubekendte. Her er ligningssystemet løst i et **Note**-værksted – først med **Enter** for at få en eksakt løsning og derefter med **Ctrl/Cmd Enter** for at få en tilnærmet løsning:

Tip
Hvis du ikke bruger skabelonen til indtastningen af ligningssystemet, skal du skrive **system(ligning1, ligning2).**



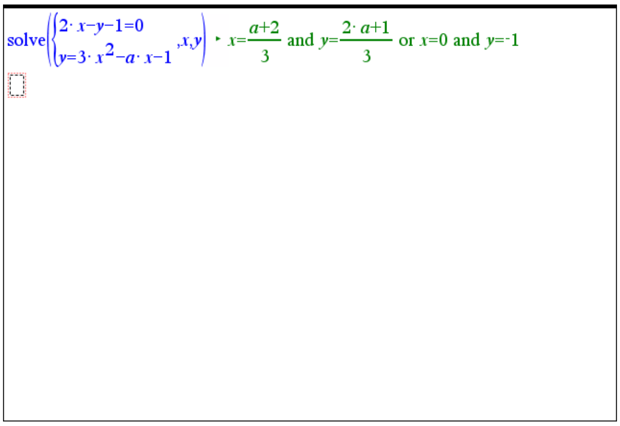
3. Ligninger med parametre

Måske har det undret dig, at du altid skal skrive, hvilken eller hvilke variable du vil løse ligningen med hensyn til. Det hænger sammen med, at TI-Nspire CAS også kan håndtere ligninger med parametre:

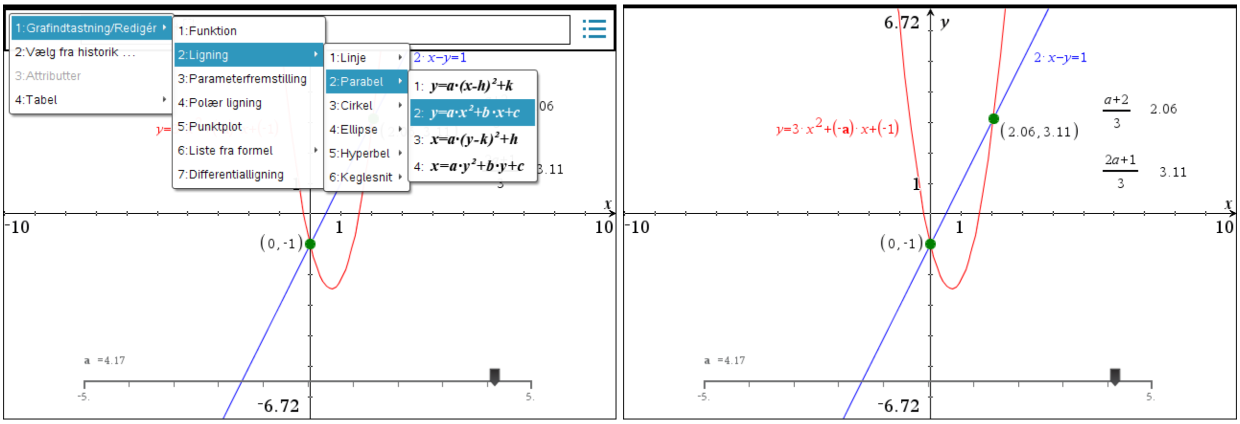
Løs ligningssystemet

$$2x - y - 1 = 0 \wedge y = 3x^2 - a \cdot x - 1$$

Geometrisk svarer denne opgave til at finde skæringspunktet mellem en ret linje og en parabel. Indtast nu ligningssystemet som vist nedenfor:



Hvis vi skal illustrere løsningen geometrisk skal vi gå lidt mere forsigtigt frem. Vi kan igen indtegne kurverne ved brug af graftypen **Ligninger** i et **Graf**-værksted, men denne gang skal vi også indføre en skyder for parameteren a. For at undgå at påvirke resultatet af den ovenstående beregning i **Note**-værkstedet er det nu afgørende at vi opretter **Graf**værkstedet i en ny opgave, så **Note**-værkstedet beholder parameteren a som en *symbolsk variabel*, mens **Graf**-værkstedet håndterer parameteren a som en *konkret numerisk variabel*.



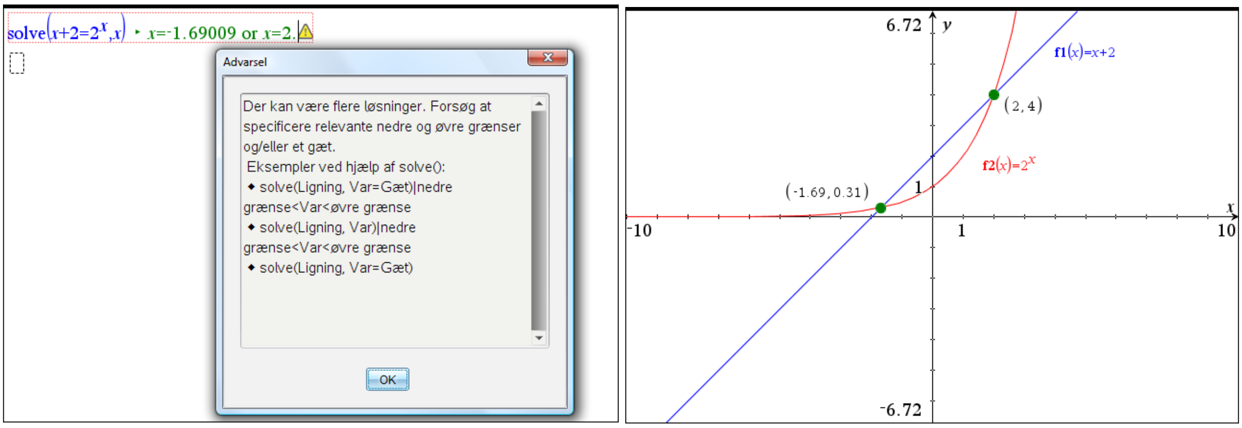
Ikke blot kan du da se, at linjen og parablen altid skærer hinanden i punktet (0,-1). Du kan også se, at det andet skæringspunkt passer med den fundne formel, idet du som vist kan indføre tekstbokse med løsningsformlerne for x og y og derefter få dem udregnet (hvor du taster L som værdi for a for at trække skyderens værdi ind i formlen).

4. Numerisk løsning af ligninger

Løs ligningen $x+2=2^x$

Tip
Du kan styre **solve** med et *gæt* på løsningen, fx solve(x+2 = 2^x, x = 1)

Hvis du løser ligningen i et **Note**-værksted, vil du se en lille advarselstrekant. Klikker du på denne, vil du se, at advarslen skyldes, at der kan være flere løsninger.



Tip
Du kan også styre **nsolve** med et gæt på løsningen. Det er specielt nyttigt, hvis **solve** helt må opgive at komme med en løsning, og du ved, at der er en.

Læg mærke til, at TI-Nspire CAS denne gang opgiver at regne symbolsk og i stedet regner videre numerisk. I den slags situationer er det specielt vigtigt at bruge **Graf**-værktøjet for at se efter, om alle løsninger nu også er fundet ☺

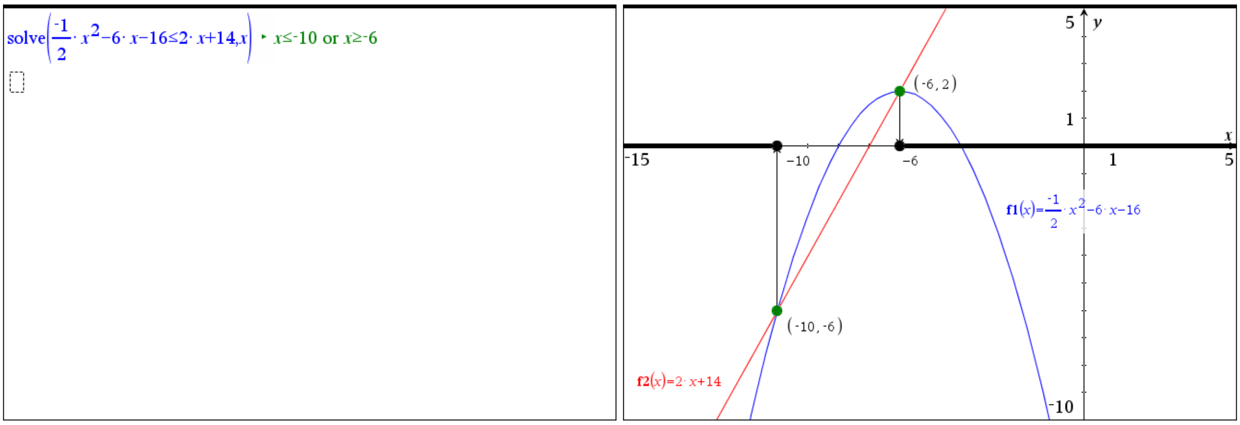
Det hænder at **solve**-kommandoen må opgive at finde en løsning fordi den kaster sig ud i alt for vilde forsøg på at finde den. Så kan man lige så godt skifte til den numeriske solver **nsolve()** og lade den prøve kræfter med ligningen.

5. Løsning af uligheder

Du kan også løse uligheder ved hjælp af **Solve**-kommandoen:

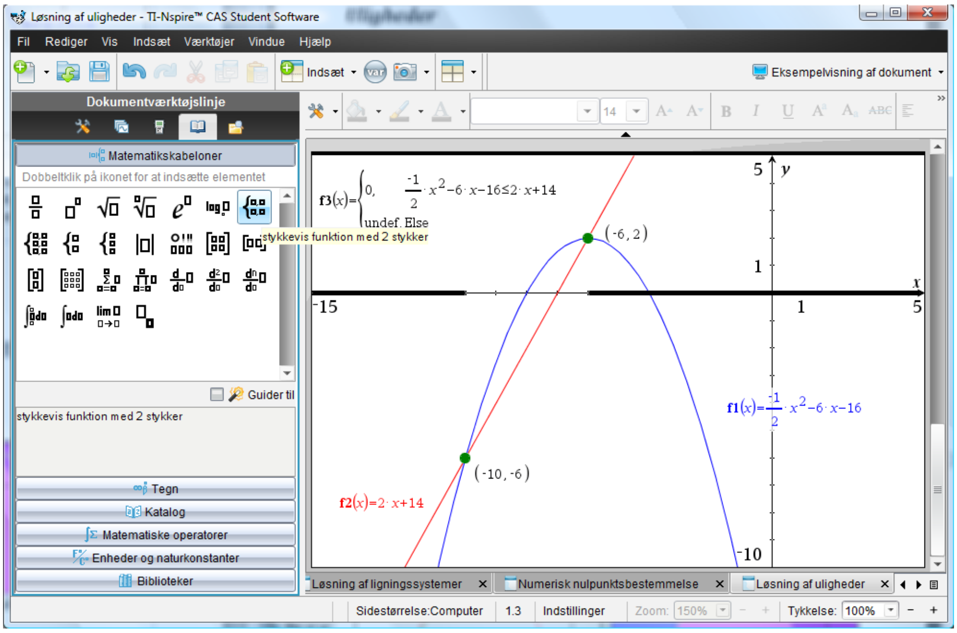
Løs uligheden $-\frac{1}{2}x^2-6x-16\leq 2x+14$

Uligheden indtastes præcis som en ligning — blot skal du anvende et ulighedstegn i stedet for et lighedstegn. Ulighedstegnet ≤ kan hentes i tegn-oversigten eller blot indskrives på formen <=. På det højre skærmbillede ser du en grafisk illustration af løsningen, hvor vi har tilføjet løsningsintervallerne med håndkraft.



Du kan imidlertid også få TI-Nspire CAS til at tegne løsningsintervallerne automatisk ved at bruge en stykvis defineret *løsningsfunktion*, som kun er defineret, når uligheden er opfyldt. Vi giver løsningsfunktionen værdien 0, så grafen/løsningsintervallerne trækkes op langs *x*-aksen. Husk at opfede grafen for løsningsfunktionen, da den ellers falder i et med *x*-aksen!

Tip
I stedet for at bruge skabelonen der godt kan drille lidt, kan du i stedet bruge **When**-kommandoen When(betingelse, sand-værdi, falsk-værdi), dvs. skrive *f3*(*x*) = when(-½·*x*²-6·*x*-16 ≤2·*x*-14 , 0, undef)



Løsning af ligninger uden brug af Solve-kommandoen

Der er mange situationer hvor det er fint nok at lade CAS-værktøjet løse ligninger med **Solve**-kommandoen. Men der er også mange situationer, hvor det er vigtigt for forståelsen at man selv kan løse simple ligninger og ligningssystemer i hånden.

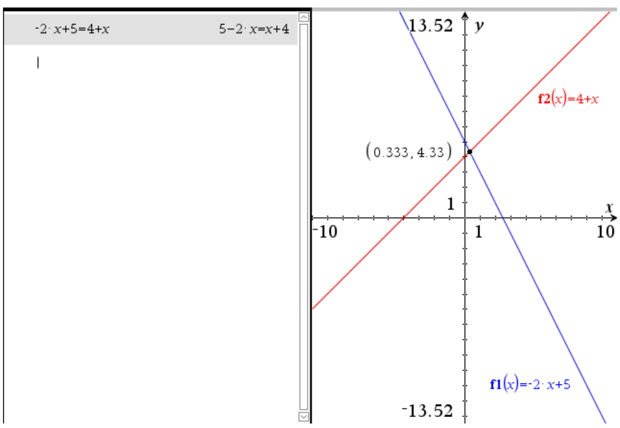
Så vi vil nu se nærmere på hvordan man kan træne sådanne færdigheder med støtte fra TI-Nspire CAS *uden* at støtte sig til **Solve**-kommandoen, men med brug af de samme teknikker, som man bruger når man arbejder med papir og blyant. TI-Nspire CAS er en fantastisk træningspartner, fordi programmet ikke bare har en engels tålmodighed men også tydeligt viser, hvornår man gør det rigtigt og hvornår man gør det forkert ☺.

1. Løsning af en simpel lineær ligning

Vi lægger ud med at dikutere hvordan man kan løse en simpel lineær ligning som fx

$$-2x+5=4+x$$

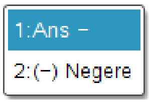
Geometrisk er der tale om skæring mellem to rette linjer, givet som grafer for henholdsvis venstresiden og højresiden af ligningen. Det er nemmest at løse ligninger i **Beregnings**-værktøjet, så vi viser først teknikken her. Vi opretter altså en delt side med et **Beregnings**-værksted til venstre og et **Graf**-værksted til højre:



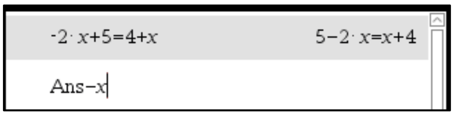
2: Ligninger og uligheder

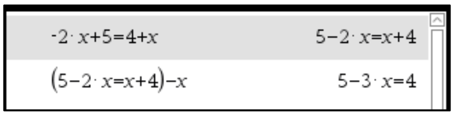
Efter at have indskrevet ligningen i **Beregnings**-værkstedet overvejer vi nu løsnings-strategien. Vi skal have samlet *x*'erne på venstre siden og konstanterne på højresiden. Vi lægger derfor *x* til på begge sider af ligningen, ligesom vi trækker 5 fra på begge sider af ligningen.

I første omgang taster vi derfor - på tastaturet. Her svarer programmet med et spørgsmål: Er det et regne -minus eller et fortegns -minus (negation):



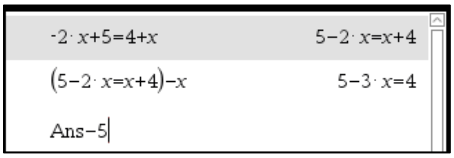
Da det er et regne-minus vælger vi den første mulighed og taster derefter *x* og **Enter**

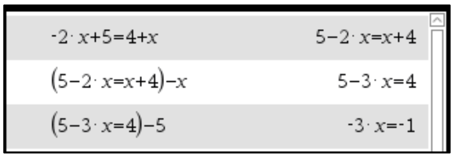




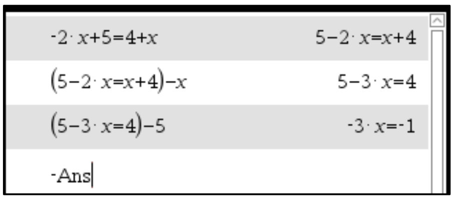
Før det afsluttende **Enter** indsættes automatisk et **Ans** (for <u>Ans</u>wer). Efter det afsluttende **Enter** erstattes **Ans** af det sidst udregnede udtryk, dvs. det reducerede ligningssystem.

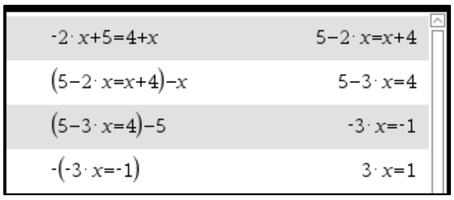
I næste trin trækker vi nu 5 fra på begge sider på samme måde ved at taste -



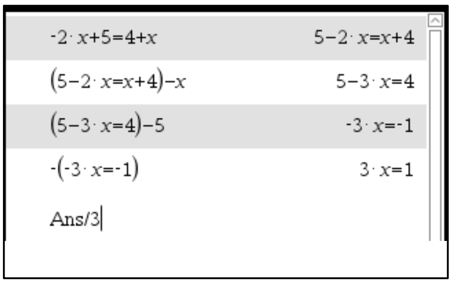


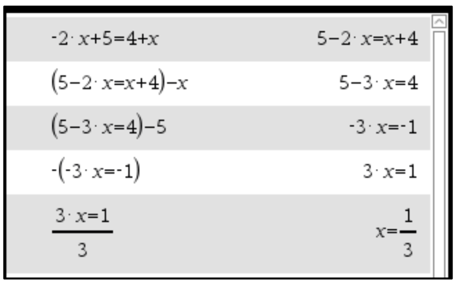
Vi har nu fået samlet *x*'erne på venstresiden og konstanterne på højresiden. Selv om det ikke er afgørende anses det for en skønhedsfejl at koefficienten til *x* er negativ. Vi skifter derfor fortegn på begge sider af ligningen ved denne gang at taste - og vælge fortegnsskiftet. Denne gang må vi selv skrive **Ans**!





Sådan! Nu ser det pænere ud. Så skal vi blot dividere ligningen med 3 på begge sider for at isolere *x*. Denne gang taster vi derfor /

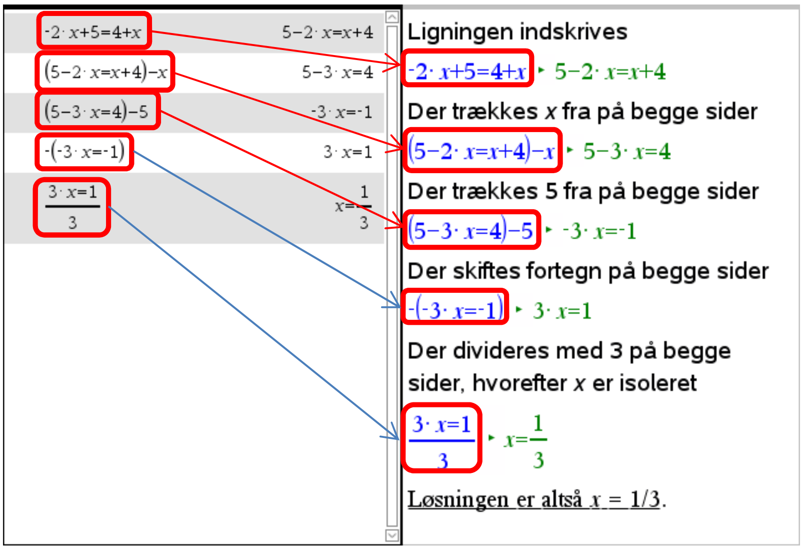




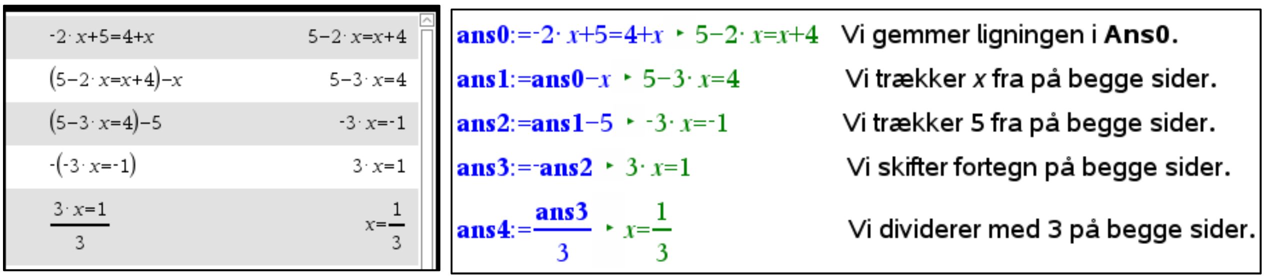
Og dermed er *x* isoleret og vi finder den samme løsning som i **Graf**-værkstedet ☺.

Kunne vi i princippet ikke have gjort det samme i et **Note**-værksted? Jo, i princippet, men ikke lige så fleksibelt, gennemskueligt og elegant ☺ **Note**-værkstedet understøtter således ikke **Ans** og man kan fx ikke lige pile op og hente tidligere brugte udtryk.

Der er nu flere strategier man kan vælge mellem: Hvis man gerne vil dokumentere løsningsmetoden skriftligt, udfører man selve løsningsmetoden i **Beregnings**-værkstedet og trækker derefter de enkelte trin over i et **Note**-værksted og tilføjer passende kommentarer, så tankegangen fremstår tydeligt. Læg mærke til at **Beregnings**-værkstedet er et beskyttet værksted, så uderegningerne forsvinder ikke fra **Beregnings**-værkstedet, når man trækker dem over ☺ Efterfølgende kan man jo så slette **Beregnings**-værkstedet, hvis det ikke længere tjener sit formål som dokumentation af løsningsmetoden.



I en lidt mere avanceret strategi, der kræver lidt fortrolighed med programmering, kan man selv indføre en *nummereret* **Ans**-variabel, der holder styr på udregningerne. I et **Note**-værksted vil den samme løsning af ligningen fx kunne se således ud:



Denne gang er alle ligningerne i **Note**-værksted kædede, så retter man i fx den første ligning slår rettelserne igennem i de følgende ligninger ☺. Til gengæld er det ikke helt så tydeligt, hvad de forskellige **ans**-variable står for – der skal man ind og nærlæse linjerne trin for trin!

Endelig skal vi bemærke at teknikken selvfølgelig også virker på rene bogstavligninger, som fx

$$a \cdot x + b = c \cdot x + d$$

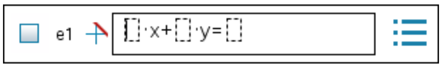
Prøv selv ☺. Undervejs i løsningsprocessen får du en advarsel. Husk at kommentere denne! Hvorfår får du en advarsel og hvad er det helt præcis vi skal tage højde for?

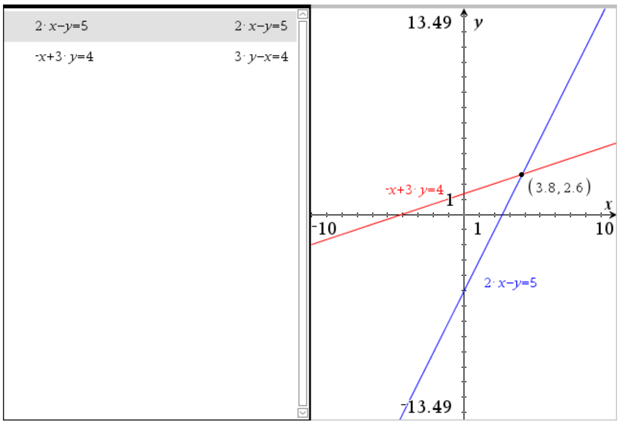
2. Løsning af to lineære ligninger med to ubekendte

De samme teknikker kan bruges til løsning af lineære ligningssystemer, fx

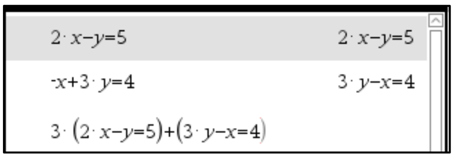
$$2x - y = 5$$
$$-x + 3y = 4$$

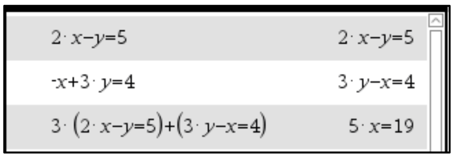
Geometrisk er der igen tale om skæring mellem to rette linjer. Linjerne tegnes nemmest ved at skifte til **Graf**-typen ligninger



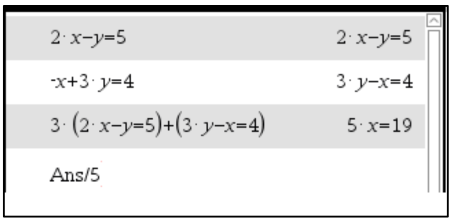


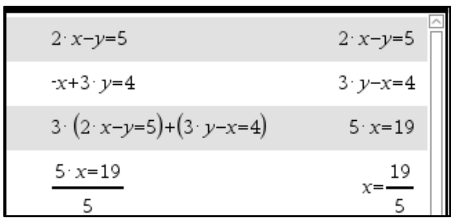
Vi vil nu løse ligningerne med lige store koefficienters metode. Vi vil da fx først finde x. Vi skal da have elmineret y. Ved at gange den øverste ligning med 3, opnår vi at de to ligningere får samme koefficient til y, dog med modsat fortegn. Derefter kan ligningerne lægges sammen og leddene med y forsvinder ud af ligningen. Da **Ans** refererer til den sidste ligning skal vi denne gang gå lidt anderledes frem. Vi taster 3 ganget med parentes, piler op og henter den første ligning og lægger en parentes til, hvorefter vi piler op henter den anden ligning:



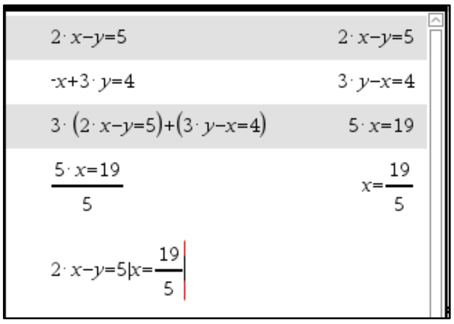


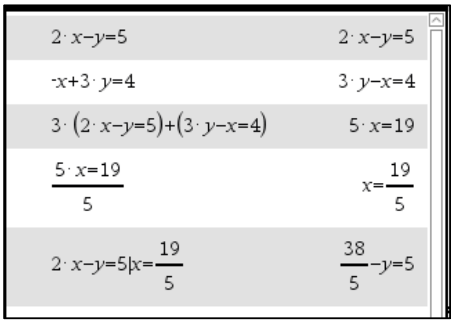
Derefter dividerer vi på begge sider med 5 og vi har isoleret x.



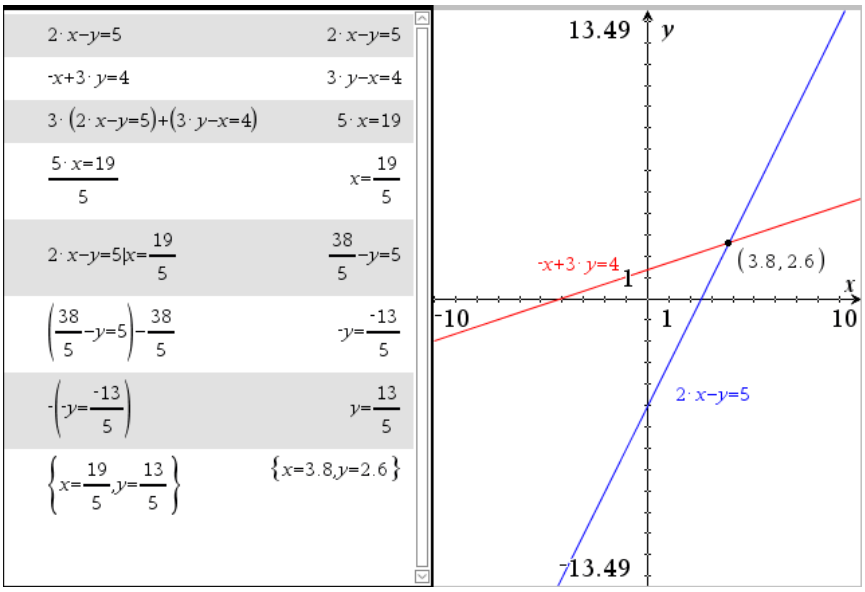


Vi kunne nu gøre det samme med y. Men ofte skifter man nu til substitutionsmetoden og indsætter den fundne værdi af x i fx den første af ligningerne. Vi piler derfor op og henter den første ligning, taster | og piler op og henter den fundne værdi af x:





Derefter isoleres y på sædvanlig vis og tilsidst piler vi op og henter de to fundne løsninger ind i en krøllet parentes – og vi taster **Ctrl/Cmd Enter** for også at se den tilnærmede løsning:



Som det ses er der fuld overensstemmelse med den grafiske løsning ☺

Igen bemærker vi, at teknikken selvfølgelig også virker på rene bogstavligninger, som fx

$$a_1 \cdot x + b_1 \cdot y = c_1$$
$$a_2 \cdot x + b_2 \cdot y = c_2$$

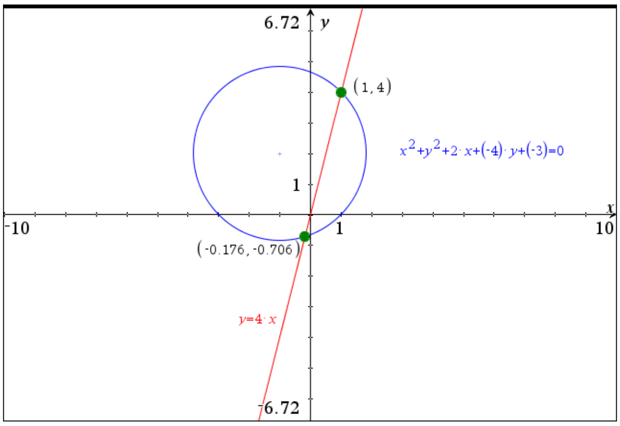
Prøv selv ☺. Undervejs i løsningsprocessen får du en advarsel. Husk at kommentere denne! Hvorfår får du en advarsel og hvad er det helt præcis vi skal tage højde for?

3. Gensyn med skæringen mellem en cirkel og en ret linje*

Vi vender nu tilbage til ligningssystemet

$$x^2 + 2x + y^2 - 4y - 3 = 0 \quad \wedge \quad y = 4x$$

Denne gang består det altså af en førstegradsligning (ret linje) og en andengradsligning (cirkel). Vi har allerede set på den grafiske repræsentation af ligningssystemet:



Vi vil nu forsøge at løse det uden brug af **Solve**-kommandoen. Vi skal da vælge en passende løsningsstrategi. Det er nemt at isolere fx y i den lineære ligning (i dette tilfælde er den faktisk isoleret på forhånd). Indsættes udtrykket for y i cirklens ligning fås en andengradsligning i x og det er den vi nu skal arbejde videre med:

$x^2+2\cdot x+y^2-4\cdot y-3=0$	$x^2+2\cdot x+y^2-4\cdot y-3=0$
$y=4\cdot x$	$y=4\cdot x$
$x^2+2\cdot x+y^2-4\cdot y-3=0 \mid y=4\cdot x$	$17\cdot x^2-14\cdot x-3=0$

Vi kunne nu selvfølgelig indføre en løsningsformel for andengradsligningen fx

$$\textbf{grad2ligning}(a,b,c):=\left\{\frac{-b-\sqrt{b^2-4\cdot a\cdot c}}{2\cdot a},\frac{-b+\sqrt{b^2-4\cdot a\cdot c}}{2\cdot a}\right\} \quad \textit{Udført}$$

Men så kunne vi vel lige så godt have brugt en solve-kommando ☺. Vi må altså i stedet tænke strategisk. Ideen bag løsningen af andengradsligning er at udføre en kvadratkomplettering. Vi skal altså lægge et passende led til vesntresiden, så den kommer til at bestå af et kvadratled (på formen x^2), et dobbelt produkt (på formen $\ldots\cdot x$) og endnu et kvadratled (uden x). Det kræver lidt øvelse, så hvis man løber sur i det kan man evt. i starten udnytte at der findes en kommando **CompleteSquare** i kateloget, der kan udføre det automatisk

CompleteSquare(*udtryk*, *var*)

Men her snyder vi ikke på vægten ☺

$x^2+2\cdot x+y^2-4\cdot y-3=0 \mid y=4\cdot x$	$17\cdot x^2-14\cdot x-3=0$
© Vi dividerer med 17 på begge sider	
$\frac{17\cdot x^2-14\cdot x-3=0}{17}$	$\frac{17\cdot x^2-14\cdot x-3}{17}=0$
© Vi slår brøken i stykker	
$\text{expand}\left(\frac{17\cdot x^2-14\cdot x-3}{17}=0\right)$	$x^2-\frac{14\cdot x}{17}-\frac{3}{17}=0$
© Vi flytter $\frac{3}{17}$ over på den anden side af lighedstegnet	
$\left(x^2-\frac{14\cdot x}{17}-\frac{3}{17}=0\right)+\frac{3}{17}$	$x^2-\frac{14\cdot x}{17}=\frac{3}{17}$
© Vi lægger nu det manglende kvadratled $\left(\frac{7}{17}\right)^2$ til på begge sider	
$\left(x^2-\frac{14\cdot x}{17}=\frac{3}{17}\right)+\left(\frac{7}{17}\right)^2$	$x^2-\frac{14\cdot x}{17}+\frac{49}{289}=\frac{100}{289}$
© Hvis alt er gået godt skal venstre siden nu være kvadratet på en toleddet størrelse ☺. Vi tjekker med en faktorisering	
$\text{factor}\left(x^2-\frac{14\cdot x}{17}+\frac{49}{289}=\frac{100}{289}\right)$	$\frac{(17\cdot x-7)^2}{289}=\frac{2^2\cdot 5^2}{17^2}$
© Det gik godt! Så vi ganger nu med 289 på begge sider	
$\left(\frac{(17\cdot x-7)^2}{289}=\frac{100}{289}\right)\cdot 289$	$(17\cdot x-7)^2=100$
© Til sidst uddrager vi kvadratroden på begge sider	
$\sqrt{(17\cdot x-7)^2=100}$	$\lvert 17\cdot x-7\rvert=10$

Vi er nu ganske tæt på! Vi skal bare ud af absolutværdifunktionen, dvs. vi er nødt til at arbejde os særskilt igennem de to tilfælde. Vi skal altså splitte ligningen i to tilfælde:

$$17\cdot x-7=\pm 10$$

© Til sidst uddrager vi kvadratroden på begge sider	
$\sqrt{(17\cdot x-7)^2=100}$	$\lvert 17\cdot x-7\rvert=10$
© Første tilfælde	
$17\cdot x-7=10$	$17\cdot x-7=10$
$(17\cdot x-7=10)+7$	$17\cdot x=17$
$\frac{17\cdot x=17}{17}$	$\boxed{x=1}$
© Andet tilfælde	
$17\cdot x-7=-10$	$17\cdot x-7=-10$
$(17\cdot x-7=-10)+7$	$17\cdot x=-3$
$\frac{17\cdot x=-3}{17}$	$\boxed{x=\frac{-3}{17}}$

Vi finder altså via lidt krøllede omskrivninger, at der er to løsninger: $x = \dfrac{-3}{17}$ og $x = 1$.

Men dem kan vi jo så sætte ind i ligningen for *y*. Vi piler altså op og henter ligningen for *y* og substituerer de fundne værdier for *x*:

$\frac{17\cdot x=17}{17}$	$x=1$
© Andet tilfælde	
$17\cdot x-7=-10$	$17\cdot x-7=-10$
$(17\cdot x-7=-10)+7$	$17\cdot x=-3$
$\frac{17\cdot x=-3}{17}$	$x=\frac{-3}{17}$
$y=4\cdot x \mid x=\frac{-3}{17}$	$y=\frac{-12}{17}$
$y=4\cdot x \mid x=1$	$y=4$
$\left\{x=\frac{-3}{17},y=\frac{-12}{17}\right\}$	$\left\{x=\frac{-3}{17},y=\frac{-12}{17}\right\}$
$\{x=1,y=4\}$	$\{x=1,y=4\}$

Dermed har vi fundet de samme løsninger som med den grafiske metode, men denne gang helt uden at bruge **Solve**-kommandoen ☺

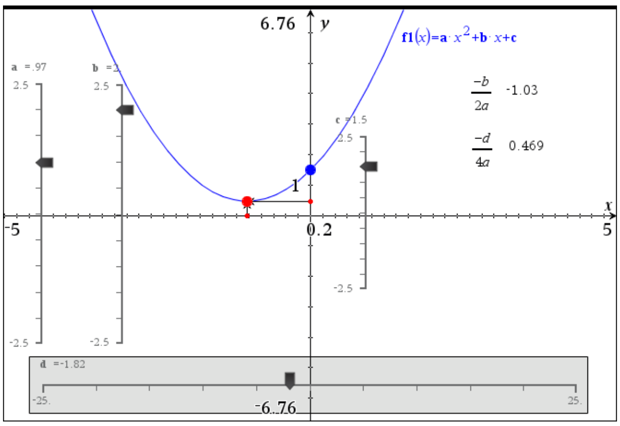
På samme måde kan vi finde skæringspunkterne mellem en parabel og en ret linje svarende til det ligningssystem vi allerede har kigget på

$$2x - y - 1 = 0 \wedge y = 3x^2 - a \cdot x - 1$$

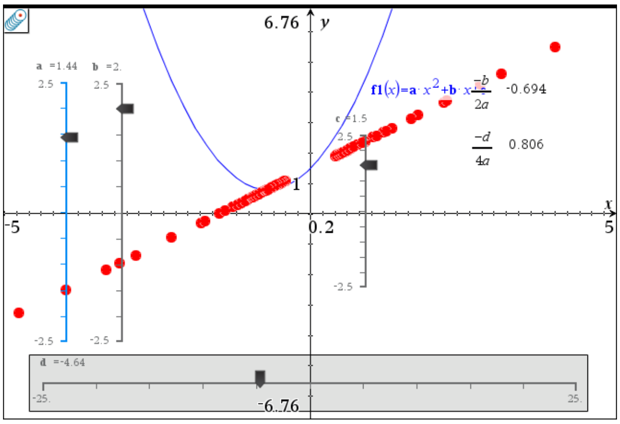
Det giver selvfølgelig en ekstra udfordring, at der nu også er en parameter *a* involveret! Prøv nu selv kræfter med dette problem ☺

*Gensyn med toppunktsformlen for en parabel**

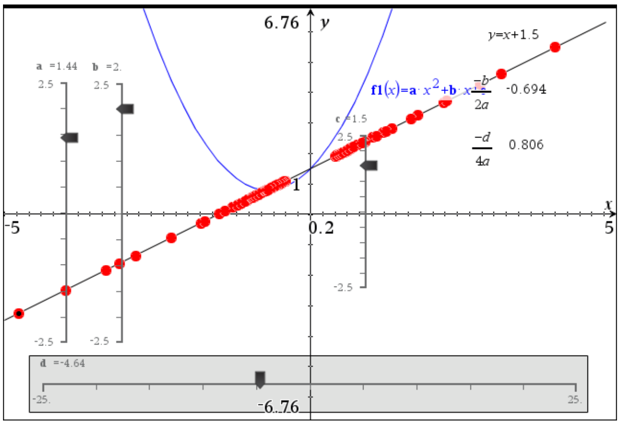
Vi slutter med et gensyn med toppunktsformlen for en parabel, som vi konstruerede en interaktiv illustration af i slutningen af det foregående kapitel. Du bør derfor arbejde med denne interaktive illustration først ☺



Vi vil nu sætte fokus på toppunktets opførsel når vi trækker i skyderne for *a*, *b* og *c*. Vi kigger i første omgang kun på variationen af koefficienten *a*. Vi sætter derfor et spor på toppunktet (fx ved at højreklikke og vælge **Geometrisk spor**)



Det kunne da godt se ud som om sporet fulgte en ret linje. Hvordan kan vi nu finde ligningen for denne rette linje? Ved at vælge konkrete pæne værdier for koefficienterne *b* og *c*, kan man godt gætte en ligning i det konkrete tilfælde. Fx kan vi jo lægge en ret linje lige oven i sporet og aflæse dens ligning:



I det konkrete tilfælde finder vi altså ligningen $y = x + 1.5$. Prøver vi flere konkrete tilfælde af kan vi nok godt gætte den generelle ligning, men det ville være rart med en symbolsk strategi. Nøglen er toppunktsligningerne

$$x = \frac{-b}{2a}$$
$$y = \frac{-d}{4a} = \frac{-(b^2 - 4a \cdot c)}{4a}$$

Her er det kun parameteren *a* vi varierer, så hvis vi kunne eliminere *a* kunne vi finde den ønskede ligning. Det kunne vi selvfølgelig gøre med en **Solve**-kommando

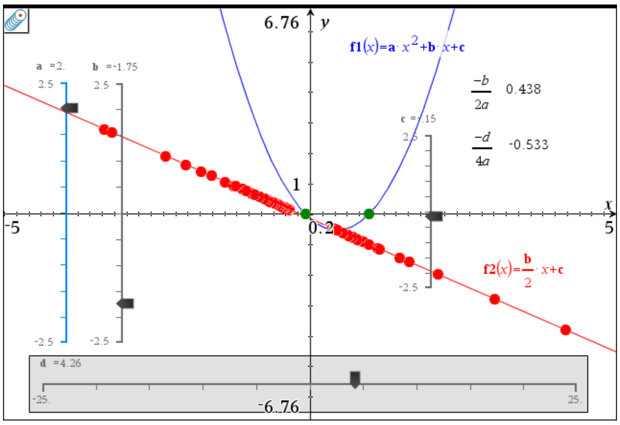
$$solve\left(\begin{cases} x = \dfrac{-b}{2a} \\ y = \dfrac{-(b^2 - 4a \cdot c)}{4a} \end{cases} \boxed{, a, y}\right)$$

Ved at løse med hensyn til a og y opnår vi netop for det første at a *ikke* optræder i løsningsformlen, for det andet at y isoleres, så vi netop får y udtrykt som en funktion af x (og parametrene b og c).

Men lad os nu først prøve om vi ikke kan gøre det selv ☺ Den første ligning er den simpleste, så vi isolerer a i den første ligning og indsætter det fundne udtryk for a i den anden ligning (husk at skifte opgave, så koefficienterne nu optræder som symbolske variable!):

$x=\frac{-b}{2\cdot a}$	$x=\frac{-b}{2\cdot a}$
$y=\frac{-(b^2-4\cdot a\cdot c)}{4\cdot a}$	$y=\frac{4\cdot a\cdot c-b^2}{4\cdot a}$
$\left(x=\frac{-b}{2\cdot a}\right)\cdot 2\cdot a$	$2\cdot a\cdot x=-b$
$\frac{2\cdot a\cdot x=-b}{2\cdot x}$	$a=\frac{-b}{2\cdot x}$
$y=\frac{4\cdot a\cdot c-b^2}{4\cdot a}\vert a=\frac{-b}{2\cdot x}$	$y=\frac{b\cdot x+2\cdot c}{2}$
$\text{expand}\left(y=\frac{b\cdot x+2\cdot c}{2}\right)$	$y=\frac{b\cdot x}{2}+c$

Den søgte ligning for sporet er altså givet ved $y=\frac{b}{2}\cdot x+c$ i fin overensstemmelse med det konkrete eksempel. Tilføjer vi grafen for den lineære funktion $y=\frac{b}{2}\cdot x+c$ ser vi da også at sporet falder på denne rette linje også for andre valg af b og c!



Herefter er turen kommet til at se hvad der sker, hvis man i stedet varierer på b, henholdsvis c, men nu er du jo fortrolig med teknikkerne, så det opfordrer vi dig til selv at udforske ☺

3 Funktioner

Du har allerede set flere eksempler på, hvordan funktioner håndteres. Hovedsagelig har du foretaget indtastningen i indtastningsfeltet i **Graf**-værkstedet og refereret til funktionerne via dets standardnavn, $f1$, $f2$, osv. Men du kan selvfølgelig også vælge helt andre navne til funktioner.

I dette afsnit vil du lære at

- definere funktioner i **Beregnings**- eller **Note**-værkstedet
- bestemme grænseværdier
- bestemme differentialkvotienter og finde tangenter
- finde stamfunktioner og benytte disse til arealbestemmelse

Funktioner i Beregnings- eller Note-værkstedet

I indtastningsfeltet i **Graf**-værkstedet *skal* du bruge x som uafhængig variabel. Definerer du en funktion i **Beregnings**-værkstedet eller i **Note**-værkstedet, er der derimod frit valg af navnet på den uafhængige variabel (input variablen). En definition af funktionen

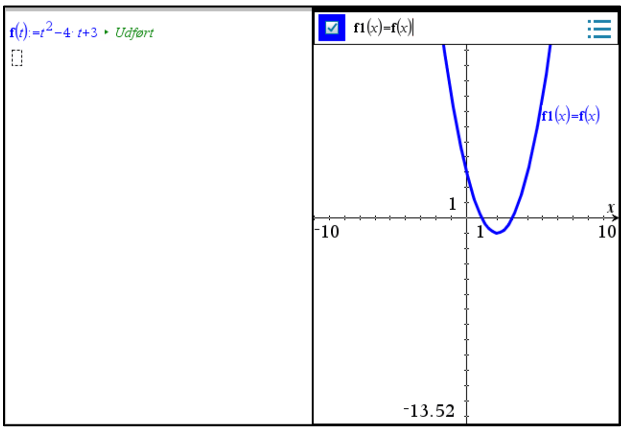
$$f(t)=t^2-4t+3$$

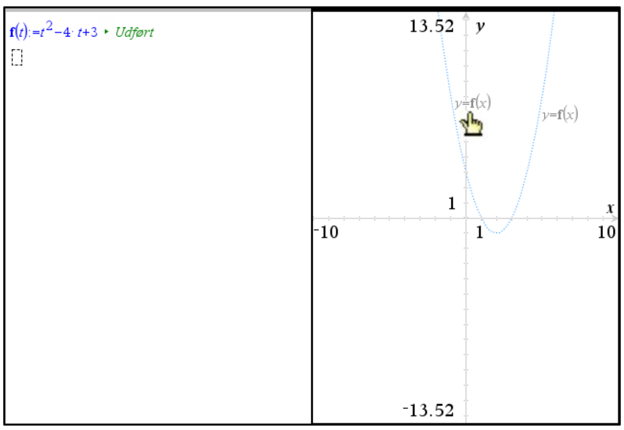
vil i **Beregnings**-værkstedet eller **Note**-værkstedet derfor se således ud:

$$f(t):=t^2-4t+3$$

Indsæt et **Note**-værksted og split skærmen i to med et **Graf**-værksted til højre. Du splitter med knappen ⊞. Husk at funktionen defineres i et matematikfelt ☺.

Grafen tegnes kan herefter tegnes i **Graf**-værkstedet ved at indtaste $f1(x) = f(x)$ i graffeltet (venstre skærmbillede) eller ved blot at taste $y=f(x)$ i et tekstfelt og derefter trække tekstfeltet ind på en af akserne (højre skærmbillede).

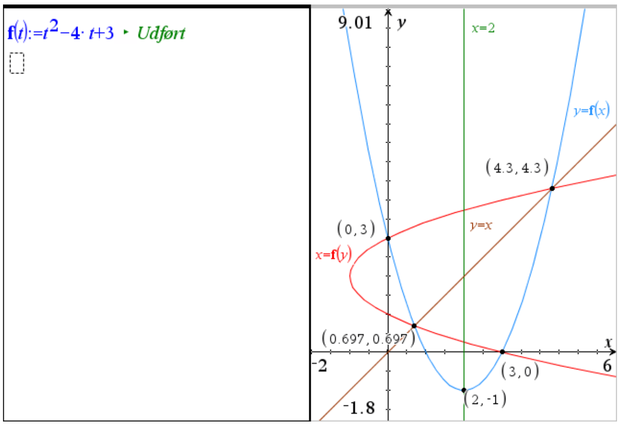




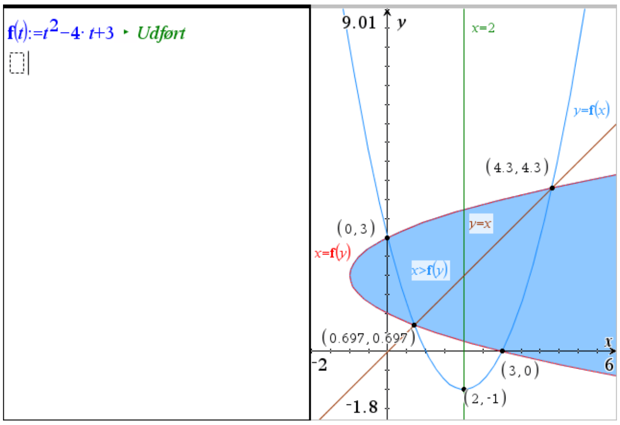
Grafen for den omvendte funktion og andre trickede grafer

TI-Nspire CAS understøtter ikke generelt omvendte funktioner. Men man kan få tegnet grafen for den omvendte funktion $y=f^{-1}(x)$ ved at skrive $x=f(y)$ i et tekstfelt og trække det ind på en af akserne. Det gælder også selv om funktionen f strengt taget ikke har en entydig omvendt funktion fordi den ikke er monoton.

På samme måde kan man få tegnet grafen for en lodret linje ved at skrive $x=værdi$ i et tekstfelt og trække det ind på en akse. Det sidste kan man dog også klare under Graftypen **Ligninger** ☺



På samme måde kan man få tegnet grafen for uligheder på formen $y > \ldots$ eller $x > \ldots$ ved at indskrive dem i tekstfelter og trække dem ind på en koordinatakse:



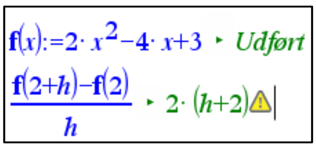
Grænseværdier

Vi ser først på en konkret grænseværdi:

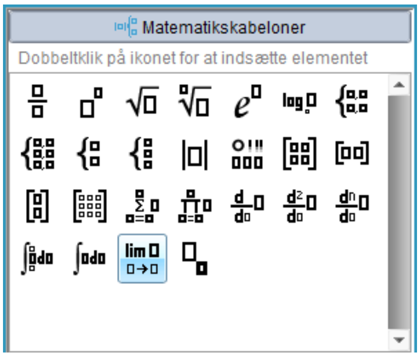
En funktion *f* er givet ved $f(x) = 2x^2 - 4x + 3$. Bestem grænseværdien af differensbrøken $\dfrac{f(2+h) - f(2)}{h}$, når *h* går mod 0.

Definer funktionen f i et matematikfelt i **Note**-værkstedet ved $f(x) := 2x^2 - 4x + 3$ og beregn differensbrøken $\dfrac{f(2+h) - f(2)}{h}$ (der angiver sekanthældningen med udgangspunkt i $x_0 = 2$ og tilvæksten *h*:

Obs
Der er en god pointe i advarslen. Differensbrøken er kun defineret for $h \neq 0$, mens det reducerede udtryk er defineret for alle h.



Grænseværdier bestemmer du ved hjælp af oversigten over matematikskabeloner



der vil indsætte grænseværdiskabelonen

Tip
I den valgfrie pladsholder skal du skrive +, hvis du vil finde grænseværdien fra højre, og –, hvis du vil finde grænseværdien fra venstre.

$$\lim_{\square\to\square}(\square)$$

En af pladsholderne i skabelonen er grå — det betyder, at det er valgfrit, om du skriver noget i denne (i almindelighed udregnes den tosidede grænseværdi, men du kan altså vælge kun at beregne grænseværdien fra den ene side).

$$f(x):=2\cdot x^2-4\cdot x+3 \;\;\blacktriangleright\;\; \textit{Udført}$$
$$\frac{f(2+h)-f(2)}{h} \;\;\blacktriangleright\;\; 2\cdot(h+2)$$
$$\lim_{h\to 0}\left(\frac{f(2+h)-f(2)}{h}\right) \;\;\blacktriangleright\;\; 4$$

Grænseværdien er altså 4.

1. Grafisk illustration af grænseværdibegrebet

Nu er grænseværdibegrebet et temmelig abstrakt begreb, så det kan være godt at få ideen illustreret lidt mere konkret, enten numerisk eller grafisk. Vi gentager derfor grænseværdi-processen i en ny opgave, med et **Note**-værksted og et **Graf**-værksted, hvor vi opretter en skyder for tilvæksten *h*, der løber fra 0 til 1 i trin af 0.001.

I **Note**-værkstedet indskrives

$$f(x):=2\cdot x^2-4\cdot x+3 \;\;\blacktriangleright\;\; \textit{Udført}$$
$$\text{dif}:=\frac{f(2+\mathbf{h})-f(2)}{\mathbf{h}} \;\;\blacktriangleright\;\; 5.6$$

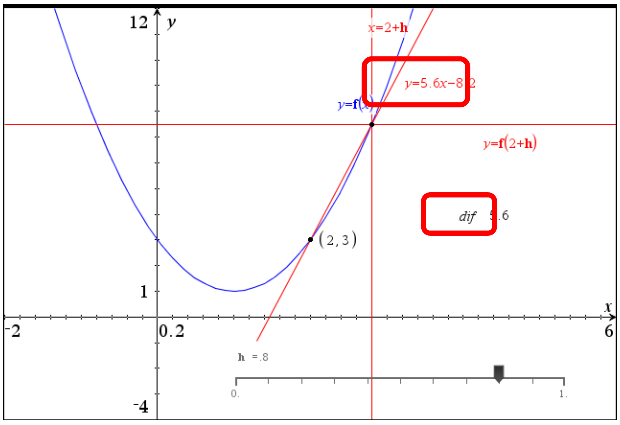
Læg mærke til at **Note**-værkstedet henter værdien for *h* i skydervariablen.

I **Graf**-værkstedet tegner vi grafen for *f*, samt sekanten hørende til $x=2$.

Sekanten er lidt tricket ☺: Punktet (2,3) afsættes som et grafpunkt, hvor vi tilretter *x*-koordinaten til 2. Punktet $(2+h, f(2+h))$ afsættes ved at indskrive tekstboksene $\boxed{x=2+h}$ og $\boxed{y=f(2+h)}$ og trække dem ind på akserne. Derved fremkommer en vandret og en lodret linje, der netop skærer hinanden i punktet $(2+h, f(2+h))$. Nu kan sekanten tegnes som en linje, der forbinder punkterne (2,3) med $(2+h, f(2+h))$ og vi kan fastlægge dens ligning.

Endelig skriver vi dif i en tekstboks, som vi beregner ved at taste L for variablen **dif**, så vi også kan se værdien af differensbrøken i **Graf**-værkstedet.

Differensbrøken angiver da netop sekanthældningen, som vi kan se af tekstboksen, der udregner differensbrøken (ligesom grænseværdien netop svarer til tangenthældningen):



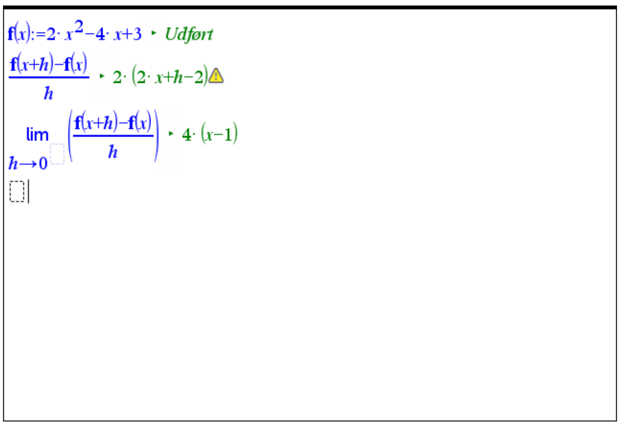
Trækker vi nu i skyderen for h, så den nærmer sig 0, kan vi nu se hvordan sekanten svinger over i tangenten og hvordan differensbrøken nærmer sig tangenthældningen 4, i overensstemmelse med at grænseværdien netop var 4.

2. Grænseovergange med funktioner

Efter dette konkrete eksempel vender vi os mod et mere abstrakt eksempel:

En funktion f er givet ved $f(x) = 2x^2 - 4x + 3$. Bestem grænseværdien af differensbrøken for f, dvs. $\dfrac{f(x+h) - f(x)}{h}$, når h går mod 0.

Definer funktionen f i et matematikfelt i **Note**-værkstedet ved $f(x) := 2x^2 - 4x + 3$, beregn differensbrøken og udregn grænseværdien som før:

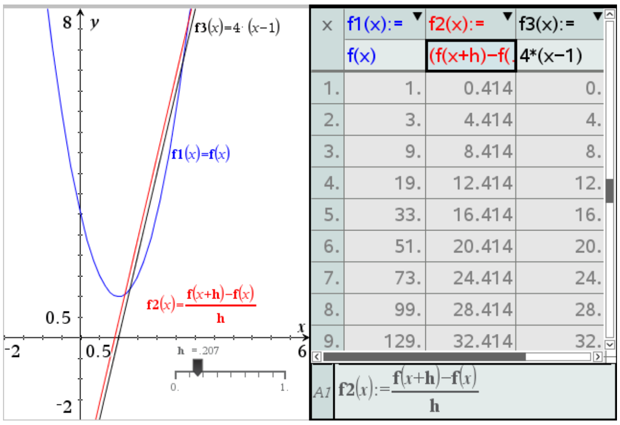


Men da grænseværdien er et temmelig abstrakt begreb, så kan det igen være godt at få ideen illustreret lidt mere konkret, enten numerisk eller grafisk. Vi opretter derfor igen en ny opgave med et **Note**-værksted og et **Graf**-værksted på hver sin side og indfører en skyder for tilvæksten h, der løber fra 0 til 1 i trin af 0.001.

I **Note**-værkstedet gentager vi så definitionen af funktionen *f* samt udregningen af differensbrøken.

I **Graf**-værksted kan vi nu tegne graferne for såvel funktionen $f(x)$, som for differensbrøken $g(x)=\dfrac{f(x+h)-f(x)}{h}$, der nærmer sig den afledede funktion $4\cdot(x-1)$, når tilvæksten *h* går mod 0.

Når vi trækker i skyderen for *h* og lader *h* nærme sig 0 ser vi da netop hvordan grafen for differensbrøken nærmer sig grafen for den afledede funktion $4\cdot(x-1)$.



Til slut går vi ind under menupunktet **Tabel** og får vist funktionstabellen. Man kan selvfølgelig også bare taste **Ctrl/Cmd T** ☺ Nu kan vi også se i tabellen, hvordan differensbrøkfunktionen nærmer sig $4\cdot(x-1)$.

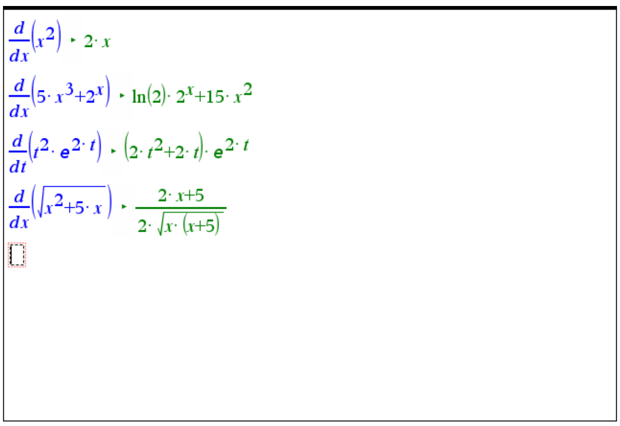
Differentialregning

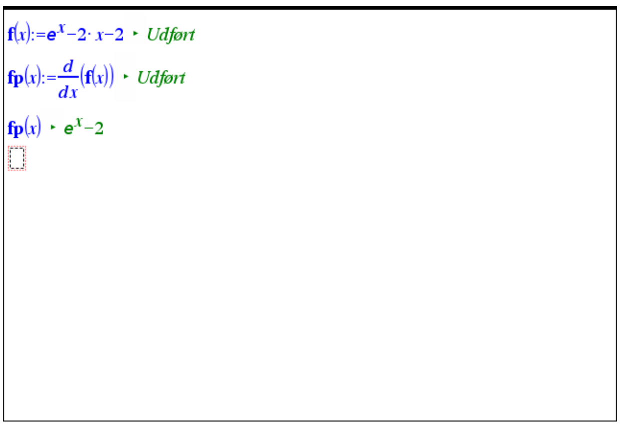
Tip
Hvis du ikke bruger skabelonen til indtastningen af differentialkvotienten, skal du skrive **derivative(expr, var)**. Du kan dog også bruge differentiationstegnet *d(* fra kataloget.

Differentialkvotienter bestemmer du vha. skabelonen $\frac{d}{d\square}\square$. Først indtaster du, hvilken variabel der skal differentieres med hensyn til, og dernæst det udtryk, der skal differentieres:

$$\frac{d}{d\square}(\square)$$

På skærmbilledet til venstre kan du se nogle eksempler (hvor grundtallet *e* for den naturlige eksponentialfunktion findes under **Tegn**-paletten eller skrives som @e).





Det er fornuftigt at definere en ny funktion *fp* ved (se højre skærmbillede)

Obs
TI-Nspire CAS understøtter *ikke* notationen $f\,'(x)$. Faktisk kan mærket ' ikke engang indgå i et variabelnavn. Det nærmeste vi kan komme er derfor betegnelser som *fp* (prime, dvs. mærke).

$$fp(x) := \frac{d}{dx}(f(x))$$

Så vil *fp* altid rumme den afledede af funktionen *f*, og *f* kan jo skifte indhold. Herefter kan du arbejde med den afledede funktion *fp* som med enhver anden funktion — herunder fx bestemme nulpunkter.

En funktion *f* er givet ved forskriften

$$f(x) = e^x - 2x - 2$$

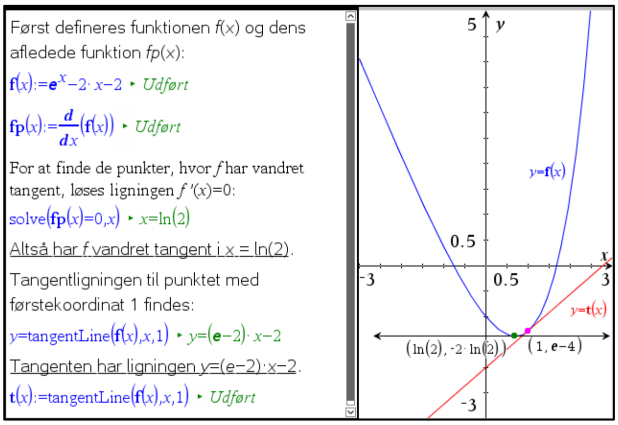
Bestem de punkter, hvor *f* har vandret tangent, og bestem tangentligningen i punktet med førstekoordinaten 1.

Første del klares ved at løse ligningen $f'(x) = 0$. Til den anden del, kan du med fordel benytte kommandoen tangentline(*f*(*x*),*x*,1), der umiddelbart giver dig et funktionsudtryk for tangenten til grafen for *f* i punktet med *x*-koordinaten 1.

Opgaven løses på en opdelt side med et **Note**-værksted til venstre og et **Graf**-værksted til højre. For bedre at udnytte pladsen er skillelinjen trukket lidt til højre.

I sidste linje er vist, hvordan tangenten kan defineres som en funktion af *x*, så den kan overføres til **graf**-værkstedet.

I **Graf**-værkstedet har vi ydermere udnyttet at vi kan arbejde med eksakte koordinater når vi fx afsætter grafpunkter ☺



1. Monotoniforhold

Vi kan også nemt fastlægge monotoniforholdene for en funktion. Som et eksempel ser vi på femtegradspolynomiet

$$f(x) = \frac{1}{15}\cdot x^5 - \frac{1}{20}\cdot x^4 - \frac{2}{3}\cdot x^3 + \frac{1}{6}\cdot x^2 + 2x + 5$$

som vi definerer i et **Note**-værksted sammen med den afledede funktion

$$f'(x) = \frac{d}{dx}f(x)$$

For at finde ud af i hvilke intervaller *f* er voksende henholdsvis aftagende skal vi nu løse ulighederne

$$f'(x) > 0 \quad \text{og} \quad f'(x) < 0$$

Først definerer vi funktionen *f* og finder den afledede funktion *fp*:

$\mathbf{f}(x) := \frac{1}{15}\cdot x^5 - \frac{1}{20}\cdot x^4 - \frac{2}{3}\cdot x^3 + \frac{1}{6}\cdot x^2 + 2\cdot x + 5$ ▸ *Udført*

$\mathbf{fp}(x) := \frac{d}{dx}(\mathbf{f}(x))$ ▸ *Udført* $\quad \mathbf{fp}(x)$ ▸ $\frac{x^4}{3} - \frac{x^3}{5} - 2\cdot x^2 + \frac{x}{3} + 2$

Så løser vi uligheden *f* '(*x*) > 0 for at finde de intervaller hvori *f* er voksende:

solve(**fp**(*x*)>0,*x*) ▸ −1.08557<*x*<1.15566 or *x*<−1.93795 or *x*>2.46786

Funktionen *f* er altså voksende i intervallerne]−∞;−1.937…[,]−1.085…;1.155…[samt]2.467…; ∞[

På samme måde finder vi de intervaller hvor f er aftagende ved at løse uligheden *f* '(*x*) < 0:

solve(**fp**(*x*)<0,*x*) ▸ −1.93795<*x*<−1.08557 or 1.15566<*x*<2.46786

dvs. *f* er aftagende i]−1.937…;−1.085…[og]1.55…;2.467…[

Det kan vi naturligvis også illustrere grafisk. Det er da bekvemt at indføre ekstremums-værdierne ved at løse ligningen $f'(x) = 0$:

top:=zeros(**fp**(*x*),*x*) ▸ {−1.93795,−1.08557,1.15566,2.46786}

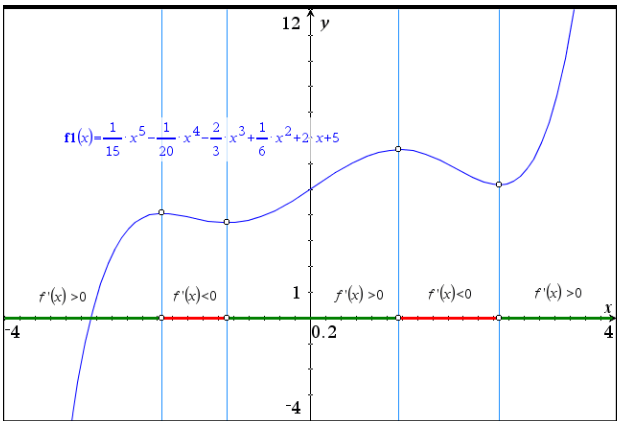
Så kan vi nemlig referere til dem som top[1],…,top[4].

I grafrummet tilføjer vi derfor tekstboksene $x = top[1]$, $x = top[2]$, $x = top[3]$ og $x = top[4]$ og trækker dem ind på akserne. Det giver anledning til fire lodrette skillelinjer, der skiller de intervaller hvor *f* er voksende fra de intervaller hvor *f* er aftagende. Vi kan nu supplere med skæringspunkterne mellem skillelinjerne og grafen for *f* (dvs. ekstremums-punkterne) samt skæringspunkterne mellem skillelinjerne og *x*-aksen.

Tilsvarende kan vi tilføje monotoniintervallerne på *x*-aksen som grafer for de betingede funktioner

$$f2(x) = when(fp(x) > 0, 0, undef) \quad \text{(Her er } f \text{ voksende)}$$
$$f3(x) = when(fp(x) < 0, 0, undef) \quad \text{(Her er } f \text{ aftagende)}$$

Graferne skal naturligvis fedes op, så vi kan se dem og de skal farves så vi kan kende forskel. Her har vi brugt grøn farve for de intervaller, hvor *f* er voksende og røde farve for de intervaller, hvor *f* er aftagende.

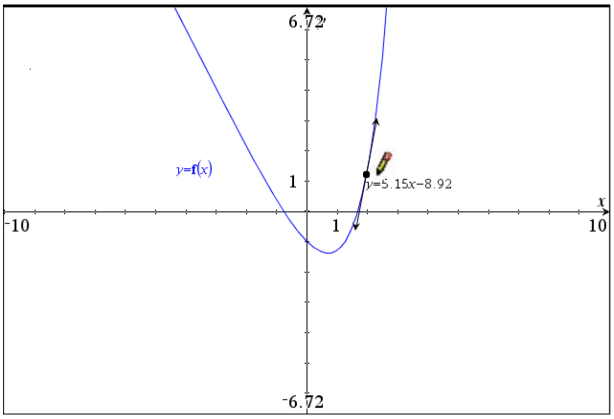


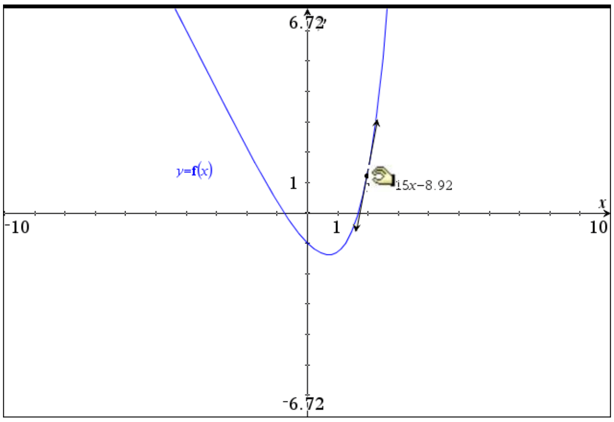
2. Dynamisk tangent

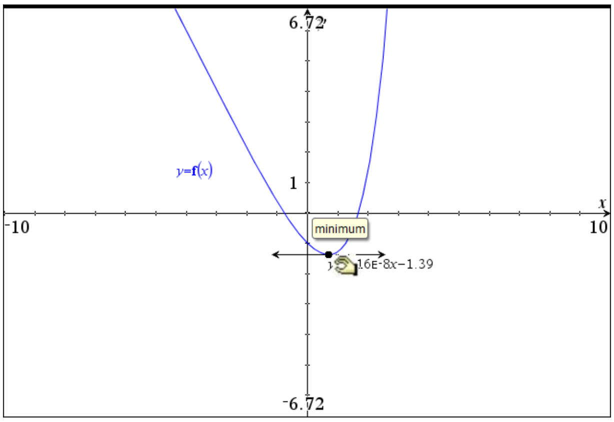
I **Graf**-værkstedet finder du et stærkt tangentværktøj, hvormed du kan klistre en dynamisk tangent på en graf. Tegn først grafen for $f(x) = e^x - 2x - 2$ i **Graf**-værkstedet. Vælg så

8:Geometri ▸ 1:Punkter og linjer ▸ 7:Tangentlinje

Klik i et punkt på grafen, og en tangent tegnes i dette punkt — skærmbilledet herunder viser situationen umiddelbart før punktet placeres. Grib punktet, og træk det langs kurven (næste skærmbillede igen):





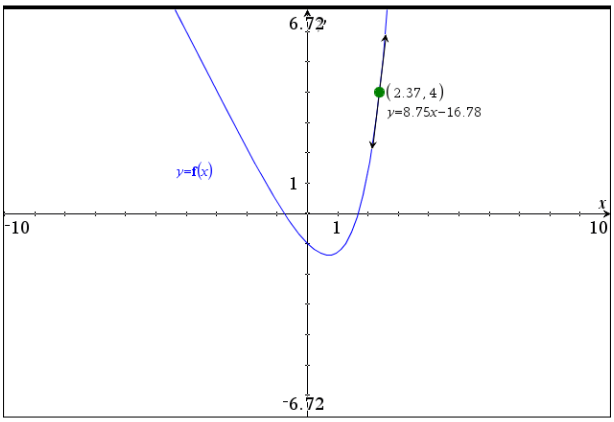


Du kan få vist røringspunktets koordinater således: Vælg

1:Handlinger ▸ 8:Koordinater og ligninger.

Klik på punktet, og flyt tangenten passende. Du kan endda springe til et bestemt punkt ved simpelthen at indtaste *x*-koordinaten (eller for den sags skyld *y*-koordinaten, som vi her har sat til 4 ☺):

Tip
Passerer du interessante punkter undervejs, så bliver du holdt underrettet.



3. Næsten-linearitet*

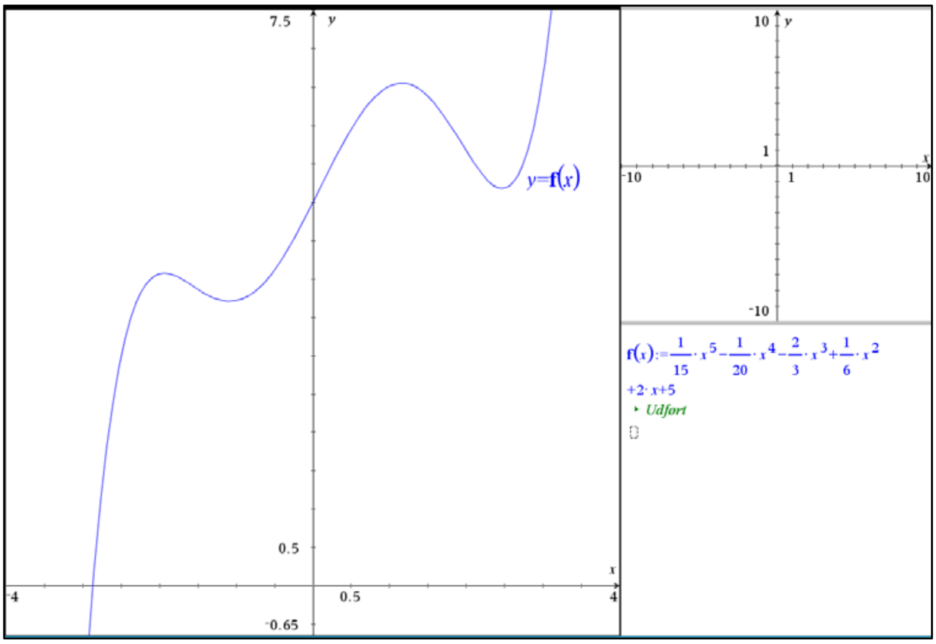
Nu hvor den dynamiske tangent er kommet på plads kan vi se lidt på mulighederne for at illustrere ideen bag differentialregning. Vi har allerede set hvordan man kan illustrere en sekant, der svinger ind til tangenten, når tilvæksten går mod 0. Vi vender os nu mod den tilsvarende illustration af en indzoomning på grafen, hvor grafen bliver mere og mere lineær for at smelte sammen med tangenten i grænsen. Det kan illustreres meget enkelt ved at vælge **Zoom Ind** i **Vindue/Zoom**-menuen og så zoome ind gentagne gange på et givet grafpunkt med tilhørende tangent.

Men man kan også opbygge en interaktiv illustration, hvor man bibeholder det oprindelige grafbillede, men omkranser grafpunktet med et zoom-kvadrat, der styres af en skyder, samtidigt med at man viser indholdet af zoom-kvadratet i et separat graf-vindue. Det er mere kompliceret, men den færdige interaktive dynamiske illustration giver et langt bedre overblik over processen, hvor man glider rundt langs grafen og vælger et sted man efterfølgende zoomer ind på og ser hvordan grafen indenfor zoomkvadratet bliver mere og mere retlinjet og til sidst smelter sammen med tangenten.

Vi skal altså have opbygget to **Graf**-værksteder, hvor det ene skal afspejle indzoomningen på det andet. Det kan da være bekvemt at lade indzoomnings-rummet være kvadratisk. Vi splitter derfor arbejdsvinduet i tre værksteder: Et stort **Graf**-værksted til venstre og et lille kvadratisk **Graf**-værksted til højre oven på et tilsvarende **Note**-værksted, hvor vi kan gemme diverse definitioner af funktioner og hjælpevariable. Det kan være lidt svært at udskille et præcist kvadratisk grafvindue, men bare det ligner så nogenlunde går det også ☺. I det følgende skærmbillede har vi afbildet femtegradspolynomiet

$$f(x) = \frac{1}{15}\cdot x^5 - \frac{1}{20}\cdot x^4 - \frac{2}{3}\cdot x^3 + \frac{1}{6}\cdot x^2 + 2x + 5$$

i et passende graf-vindue: $-4 < x < 4$ og $-0.65 < y < 7.50$. Der arbejdes på en stor skærm, så man bedre kan se detaljerne ☺

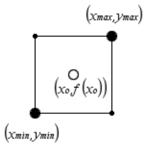


Vi skal så have indført et punkt på grafen som vi kan trække rundt i og en skyder for tilvæksten h, der som sædvanlig går fra 0 til 1 i trin af 0.001.

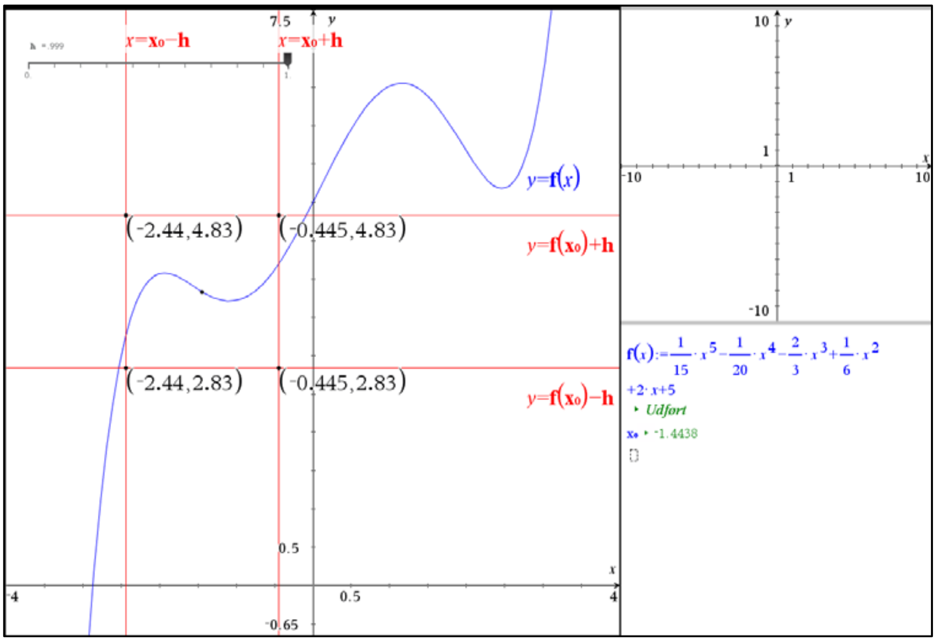
Førstekoordinaten til grafpunktet lagres i variablen x_0 (ved at højreklikke på koordinaten)

og derefter indskrives tekstboksene $\boxed{x = x_0 - h}$, $\boxed{x = x_0 + h}$, $\boxed{y = f(x_0) - h}$ samt $\boxed{y = f(x_0) + h}$. Læg mærke til at vi trækker *h* fra $f(x_0)$ ligesom vi lægger *h* til $f(x_0)$.
Vi er altså i færd med at omkranse grafpunktet med et *kvadrat* med centrum i grafpunktet og 'radius' *h*.

Disse fire tekstbokse trækkes nu ind på en koordinatakse og vi har konstrueret skelettet til zoomkvadratet. For at finde hjørnerne i zoomkvadratet konstruerer vi skæringspunkterne.



Derefter lagres koordinaterne til to af hjørnerne, der ligger diagonalt overfor hinanden i variablene $x_{\min}$, $y_{\min}$ samt $x_{\max}$, $y_{\max}$. Vi kan nu skjule hjælpelinjerne og koordinaterne og trække zoomkvadratet op som en polygon, der fedes passende op og farves grøn i kanten og lysegrå i det indre. Vi klistrer også en dynamisk tangent fast til grafpunktet.

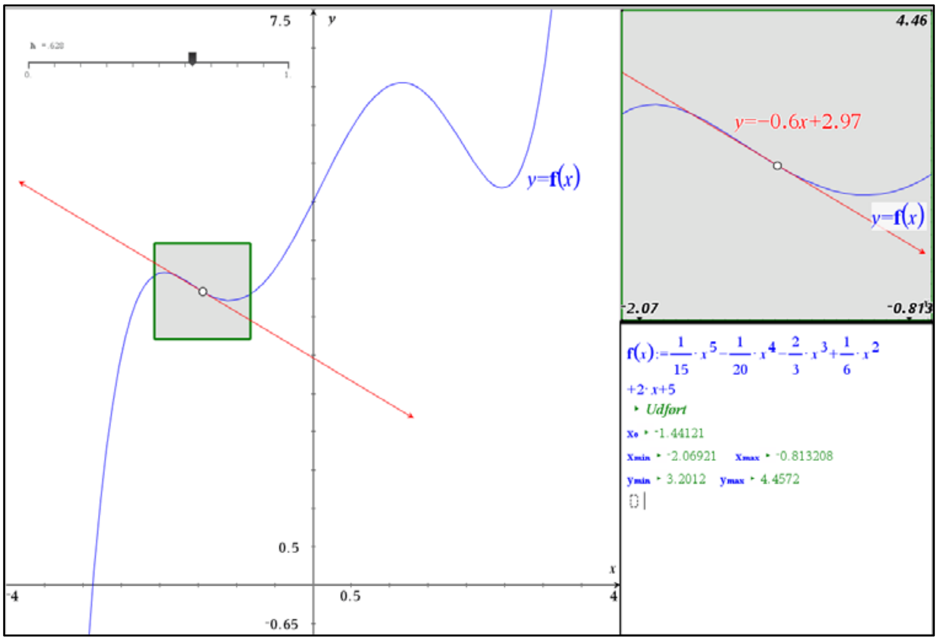


Herefter vender vi os mod det kvadratiske **Graf**-værksted, hvor vi kæder aksernes endeværdier til variablene $x_{\min}$, $y_{\min}$, $x_{\max}$ og $y_{\max}$ så det kvadratiske grafvindue netop afspejler zoomkvadratet. Vi kan så tænde for grafen for *f* og tilføje et frit punkt med en dynamisk tangent klistret fast til grafpunktet.

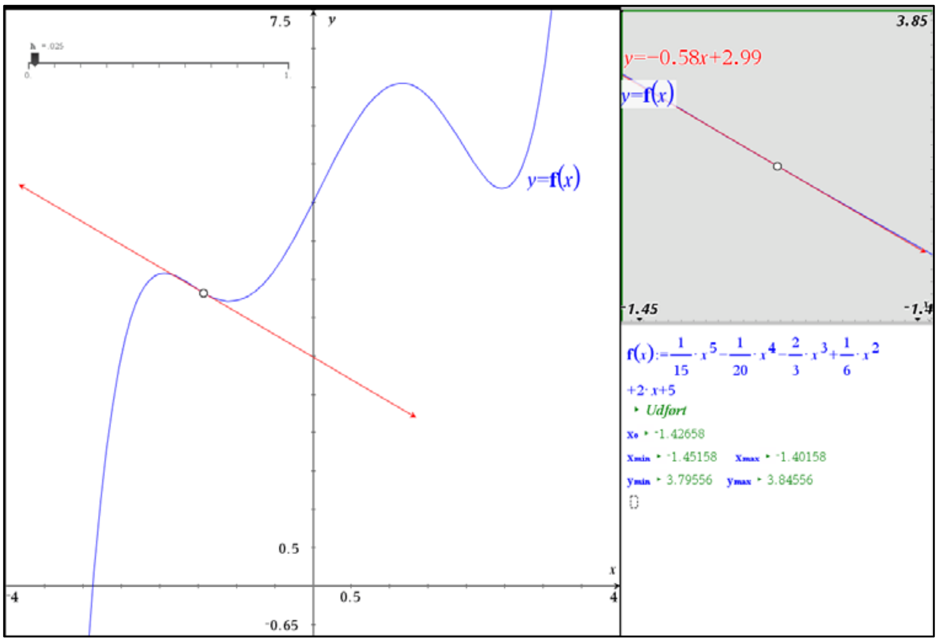
Vi kæder så *x*-koordinaten for dette frie grafpunkt til variablen x_0 (ved at højreklikke på x-koordinaten og vælge **Variable** ▸ **Kæd til**), så det afspejler det samme punkt som i det store **Graf**-værksted.

Endelig kan vi trække et kvadrat, der følger randen af det lille graf-vindue, trække kanten op i grønt og det indre i lysegråt efter præcis samme opskrift som i det store graf-vindue, dvs. vi indskriver tekstboksene $\boxed{x = x_0 - h}$, $\boxed{x = x_0 + h}$, $\boxed{y = f(x_0) - h}$ samt $\boxed{y = f(x_0) + h}$.
Disse fire tekstbokse trækkes derefter ind på en koordinatakse og vi har konstrueret skelettet til kvadratet. For at finde hjørnerne i kvadratet konstruerer vi tilsidst skæringspunkterne.

Dette kvadrat følger nu med ved indzoomningen præcis ligesom aksernes endeværdier gør det ☺



Så er vi sådan set færdige. Tilbage er blot at trække i skyderen og se hvordan grafen og tangenten smelter samme inde i zoomkvadratet. Samtidigt kan man selvfølgelig også trække i grafpunktet, så man kan se de smelter sammen overalt langs grafen.

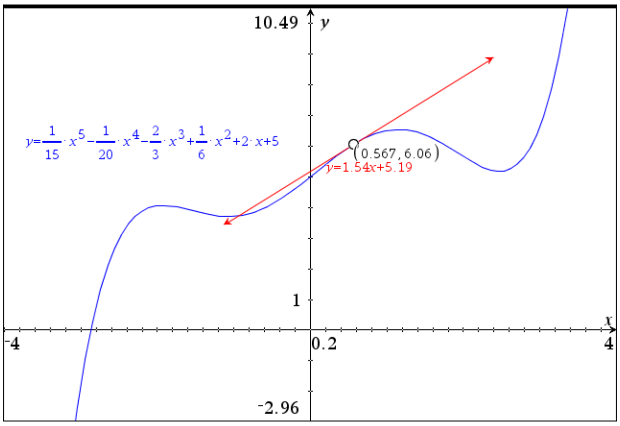


4. Grafen for den afledede funktion*

Så er det heldigvis noget nemmere at opbygge grafen for den afledede funktion f' ud fra den dynamiske tangent. Her skal vi bare udnytte at $f'(x)$ netop svarer til tangenthældningen i *x*. Det er afgørende for eksemplet at det kan gennemføres *før* man ved hvordan man differentierer symbolsk ☺. Vi tager igen udgangspunkt i femtegradspolynomiet

$$f(x) = \frac{1}{15}\cdot x^5 - \frac{1}{20}\cdot x^4 - \frac{2}{3}\cdot x^3 + \frac{1}{6}\cdot x^2 + 2x + 5$$

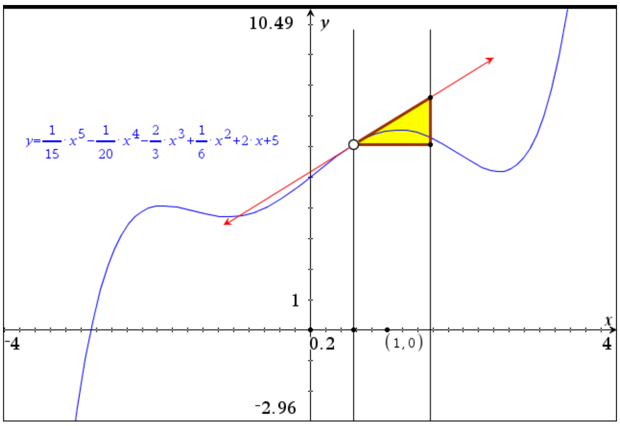
Der afsættes et frit punkt på grafen og vi klistrer en dynamisk tangent fast til grafpunktet.



Vi skal så have fat i tangenthældningen. Det kan gøres på flere måder. Her vælger vi at konstruere hældningstrekanten med den vandrette side 1 og den lodrette side svarende til hældningen rent geometrisk. Det vil tillade os senere at udnytte værktøjet **geometrisk sted**. Vi kommer derfor til at støtte os kraftigt til parallelforskydninger i det følgende! Som udgangspunkt skal vi har konstrueret hældningstrekanten ud fra grafpunktet, hvor vi altså går 1 hen og tangenthældningen op (så man lander på tangenten).

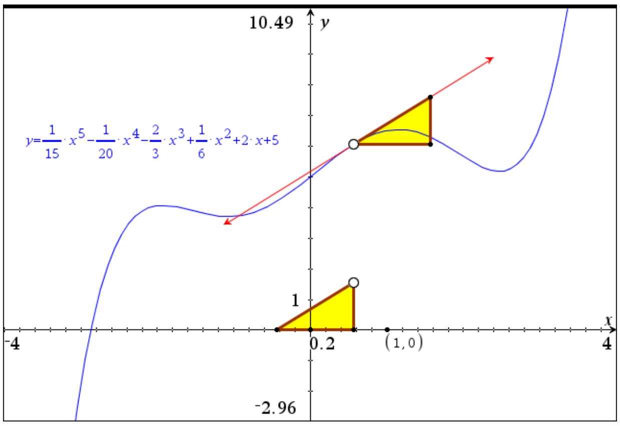
Tip
For at udføre en parallelforskydning kan man fx gå således frem med parallelforskydningsværktøjet i **Transformations**-menuen under **Geometri**: Først klikker man på start og slutpunkt for forskydningsstykket, derefter på det objekt, der skal forskydes.

Vi trækker derfor den vinkelrette fra grafpunktet til x-aksen og konstruerer fodpunktet. Tilsvarende afsætter vi begyndelsespunktet (0,0) og enhedspunktet (1,0) på x-aksen. Herefter kan hældningstrekanten konstrueres via passende parallelforskydninger. Først forskydes grafpunktet det vandrette stykke 1, derefter trækker vi den vinkelrette linje på x-aksen gennem det forskudte punkt og endelig konstruerer vi skæringen med tangenten. Hældningstrekanten kan nu trækkes op som vist

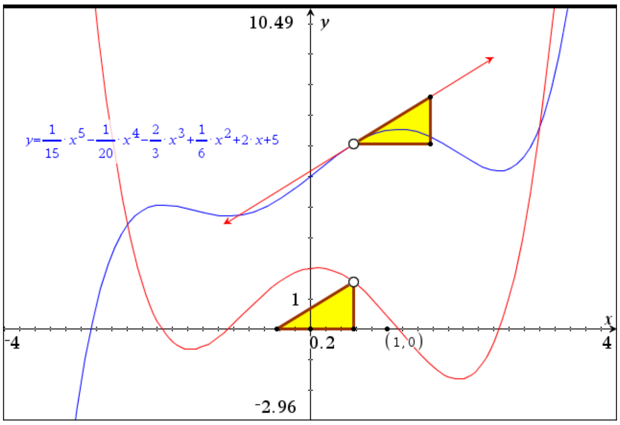


Vi skjuler hjælpelinjerne og flytter nu hældningstrekanten ned på x-aksen ved hjælp af passende parallelforskydninger. Det er afgørende, at den forskudte trekant har en lodret side der netop falder sammen med $x = x_0$, så den netop slutter i samme højde som tangenthældningen ☺ . Det øverste hjørnepunkt for den forskudte hældningstrekant spiller altså netop rollen som grafpunktet $(x_0, f'(x_0))$ for den afledede funktion.

Vi forskyder derfor dels fodpunktet lodret op det samme stykke som i trekanten, dels vandret bagud stykket 1 (klik først i de to punkter, der fastlægger parallelforskydningen og derefter i fodpunktet på x-aksen). Kontrollér, at du kan trække grafpunktet rundt langs grafen og at de to hældningstrekanter følger med som forventet ☺



Tilbage er der så blot at spore grafpunktet for den afledede funktion (i hældningstrekanten på *x*-aksen). Men fordi vi har været så omhyggelige at konstruere grafpunktet rent geometrisk, kan vi nu konstruere sporet som et *geometrisk sted* ved at udpege først grafpunktet på grafen for *f* som det drivende punkt, og derefter grafpunktet for den afledede funktion (toppunktet for hældningstrekanten langs *x*-aksen) som det punkt, der frembringer det geometriske sted:



Faktisk er det ikke helt uoverkommeligt at gætte ligningen for den afledede funktion (rød graf) fx via en passende regressionsmodel – men vi kunne selvfølgelig også bare have differentieret den oprindelige funktion: Vi overlader detaljerne til læseren ☺

Integralregning

Stamfunktioner bestemmer du ved at benytte skabelonen . Først indtaster du, hvilket udtryk der skal integreres og dernæst den variabel der skal integreres med hensyn til

$$\int \square \, d\square$$

Her ser du nogle eksempler:

$$\int (x^2+3\cdot x-7)\,dx \;▸\; \frac{x^3}{3}+\frac{3\cdot x^2}{2}-7\cdot x$$

$$\int e^{k\cdot x}\,dx \;▸\; \frac{e^{k\cdot x}}{k}$$

$$\int \frac{1}{\sqrt{x}}\,dx \;▸\; 2\cdot\sqrt{x}$$

$$\int \left(\frac{2}{x}+\frac{x}{2}\right)dx \;▸\; 2\cdot \ln(|x|)+\frac{x^2}{4}$$

Obs
Ved at bruge integraltegnet ∫(fra kataloget kan du få integrationskonstanten med. Syntaksen er
∫(*f* (*x*), *x*, *k*).
Du kan også skrive kommandoen direkte således:
Integral(*f*(*x*), *x*, *k*).

Bemærkning: Da integralregning er det modsatte af differentialregning kan du faktisk også finde stamfunktionerne ved at differentiere baglæns

Tip
For at differentiere baglæns skal du bruge skabelonen eller tilføje en ekstra parameter – til **Derivative**-kommandoen

$$\frac{d^{-1}}{dx^{-1}}\left(x^2+3\cdot x-7\right) \blacktriangleright \frac{x^3}{3}+\frac{3\cdot x^2}{2}-7\cdot x$$

$$\frac{d^{-1}}{dx^{-1}}\left(e^{k\cdot x}\right) \blacktriangleright \frac{e^{k\cdot x}}{k}$$

$$\frac{d^{-1}}{dx^{-1}}\left(\frac{1}{\sqrt{x}}\right) \blacktriangleright 2\cdot\sqrt{x}$$

$$\frac{d^{-1}}{dx^{-1}}\left(\frac{2}{x}+\frac{x}{2}\right) \blacktriangleright 2\cdot\ln(|x|)+\frac{x^2}{4}$$

1. Stamfunktionsbestemmelse

En funktion *f* er bestemt ved

$$f(x)=3x^2-21x+30$$

Bestem den stamfunktion til *f*, der går gennem punktet *P*(2,42).

Start med at definere *f*. Du kan ikke benytte navnet *F* for en stamfunktion til *f*, da TI-Nspire CAS ikke skelner mellem store og små bogstaver i variabelnavne. Brug fx navnet *sf* i stedet.

Når du bruger skabelonen til at bestemme stamfunktioner, får du ikke en integrationskonstant med i resultatet — den må du selv tilføje. Du skal derfor definere stamfunktionen ved:

$$sf(x)=\int f(x)\,dx+k$$

Opgaven er løst på nedenstående skærmbillede

Først defineres funktionen *f*(*x*) og dens tilhørende stamfunktion *sf*(*x*):

$\text{f}(x):=3\cdot x^2-21\cdot x+30 \blacktriangleright$ *Udført*

$\text{sf}(x):=\int \text{f}(x)\,\text{d}x+k \blacktriangleright$ *Udført*

$\text{sf}(x) \blacktriangleright x^3-\frac{21\cdot x^2}{2}+30\cdot x+k$

Så fastlægges integrationskonstanten ved at løse ligningen F(2) = 42:

$\text{solve}(\text{sf}(2)=42,k) \blacktriangleright k=16$

Den stamfunktion til *f*, der går gennem punktet *P*(2,42) har derfor forskriften

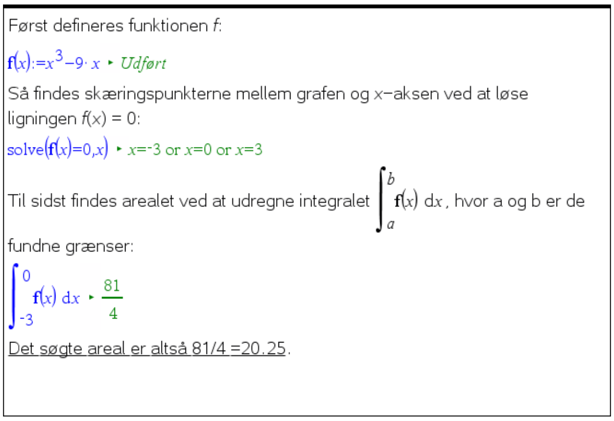
$F(x)=x^3-\frac{21\cdot x^2}{2}+30\cdot x+16$.

2. Det bestemt integral og arealbestemmelse

Til det bestemte integral benytter du skabelonen . Du udfylder som ved det ubestemte integral — blot skal du her medtage grænser. Husk, at du navigerer mellem pladsholderne med TAB (eller piletasterne):

$$\int_{\square}^{\square} \square \,\text{d}\square$$

Grafen for $f(x) = x^3 - 9x$ afgrænser sammen med x-aksen i anden kvadrant en punktmængde. Bestem arealet af denne punktmængde.


Først defineres funktionen f:

$f(x):=x^3-9\cdot x$ ▸ *Udført*

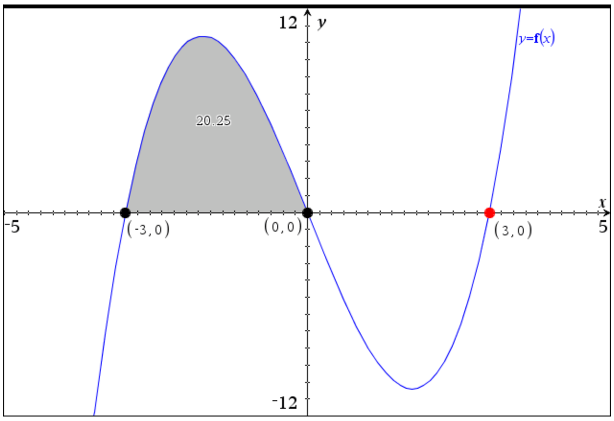
Så findes skæringspunkterne mellem grafen og x−aksen ved at løse ligningen $f(x) = 0$:

solve$(f(x)=0,x)$ ▸ $x=-3$ or $x=0$ or $x=3$

Til sidst findes arealet ved at udregne integralet $\int_a^b f(x)\,dx$, hvor a og b er de fundne grænser:

$\int_{-3}^{0} f(x)\,dx$ ▸ $\frac{81}{4}$

Det søgte areal er altså 81/4 =20.25.



Tip
Bestem først skæringspunkterne med x-aksen ved at bruge geometriværktøjet til bestemmelse af skæringspunkter (udpeg x-aksen som det ene objekt).

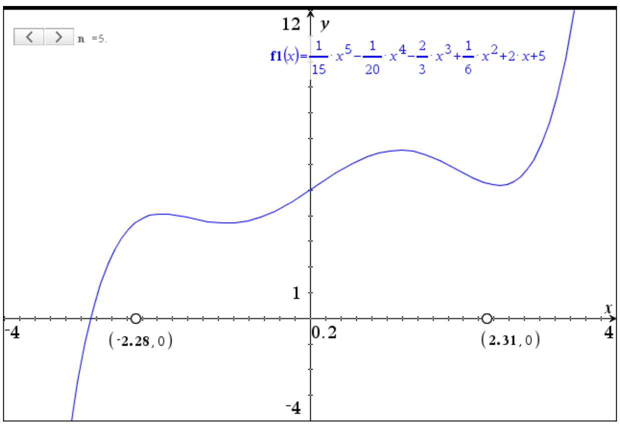
På det højre skærmbillede ser du opgaven løst med
6:Undersøg grafer ▸ 7:Integral.
Med dette værktøj skal du først udpege grænserne, dvs. de to skæringspunkter med x-aksen et efter et — du kan også indtaste grænserne direkte. Husk at sætte antallet af decimaler for værdien af integralet, da det ellers vises som 20.3 ☺. Du sætter antal decimaler op ved at højreklikke og vælge **attributter**.

3. Integralet som en sum*

Det sidste emne vi vil se på i dette kapitel er hvordan man kan illustrere integralet som en sum. Vi vil benytte en venstresum til illustrationen, men den kan sagtens udbygges til også at vise andre former for summer. Vi tager endnu engang udgangspunkt i grafen for femtegradspolynomiet

$$f(x) = \frac{1}{15}\cdot x^5 - \frac{1}{20}\cdot x^4 - \frac{2}{3}\cdot x^3 + \frac{1}{6}\cdot x^2 + 2x + 5$$

I grafvinduet skal vi nu udpege to punkter på x-aksen, der skal fungere som grænser for arealet. Vi bestemmer deres koordinater og lagrer førstekoordinaterne i variablene a og b. Endelig indfører vi en heltallig skyder n, der antager værdierne fra 1 til 20 i spring af 1. Den skal styre hvor mange delintervaller, vi opdeler integrationsintervallet $[a;b]$ i. Da det er en diskret skyder med heltallige værdier minimerer vi den.



Vi skal nu have konstrueret delintervallerne. Da antallet af delintervaller er dynamisk konstruerer vi dem ud fra et **Lister og Regneark**-værksted. I dette regneark indfører vi listerne **x_int**, **y_int** og **nul**, der står for intervalendepunkternes *x*-værdier, deres *y*-værdier samt en hjælpeliste fyldt med nuller. Listerne opbygges ved hjælp af den såkaldte **Seq**-kommando (for Sequence), der som udgangspunkt tager en formel med en indeksvariabel, som gennemløber en række indeks-værdier.

Intervaltilvæksten Δx er givet ved $\frac{b-a}{n}$. De tre formler ser nu således ud

$$x\_\text{int} := seq\left(a+\frac{b-a}{n}\cdot i, i, 0, n\right)\text{, der udregner listen } \left\{a, a+\frac{b-a}{n}, a+\frac{b-a}{n}\cdot 2, \dots, b\right\}$$

$$y\_\text{int} := f1(x\_\text{int})\text{, der udregner listen } \left\{f1(a), f1\left(a+\frac{b-a}{n}\right), \dots, f1(b)\right\}$$

$$nul := seq(0, i, 0, n)\text{, der udregner listen } \{0, 0, 0, \dots, 0\}$$

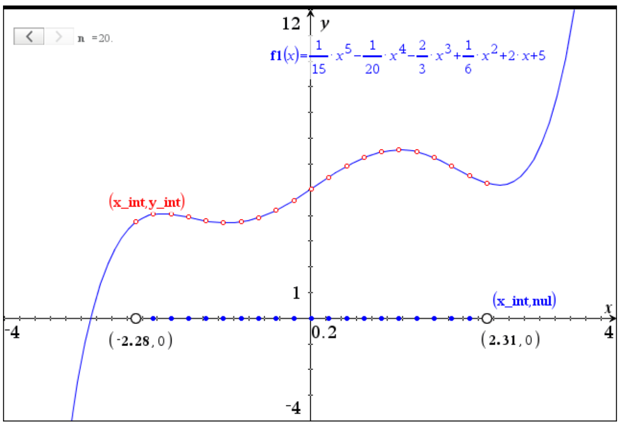
	A x_int	B y_int	C nul	D	E
=	=seq('a+('b−'a)/'n*i,i,0,'n)	=f1(x_int)	=seq(0,i,0,'n)		
1	-2.28213	3.74449	0		
2	-1.36426	3.7862	0		
3	-0.446395	4.19656	0		
4	0.471473	5.90921	0		
5	1.38934	6.47134	0		
6	2.30721	5.25551	0		
7					
8					
9					

A $x\_\text{int}:=seq\left('a+\frac{'b-'a}{'n}\cdot i, i, 0, 'n\right)$

Vi skal nu have tegnet de tilhørende rektangler og skal derfor overføre regnearkets lister til to passende punktplot:

(x_int, **nul)** og **(x_int**, **y_int)**

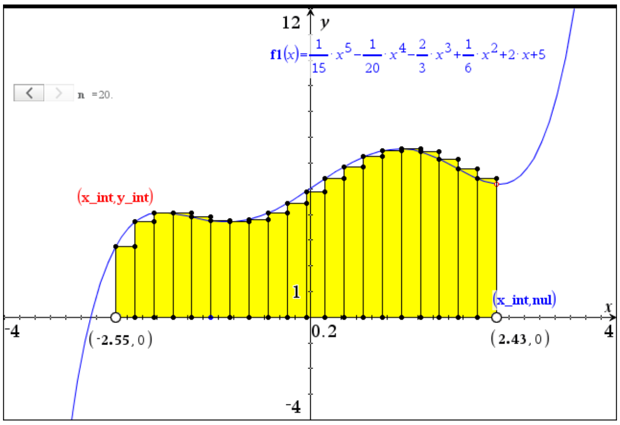
Herefter er det nu afgørende at vi skruer op for skyderen til dens maksimale værdi, så vi får tegnet alle rektanglerne!



Tip
Driller konstruktionen for meget kan man med fordel skifte funktionen midlertidig ud med en passende lineær funktion.

Vi trækker nu lodrette linjer gennem skillepunkterne og tilsvarende vandrette linjer gennem grafpunkterne. Herefter konstruerer vi skæringspunkterne hørende til venstresummen. Det kræver betydelig tålmodighed og man skal nok konstruere små portioner ad gangen og undervejs trække i grænsepunkterne for at få skilt punkterne tydeligt ad: Hvis de ligger i samme højde har de tendens til at falde sammen, så man ikke får klikket på de rette punkter ☺.

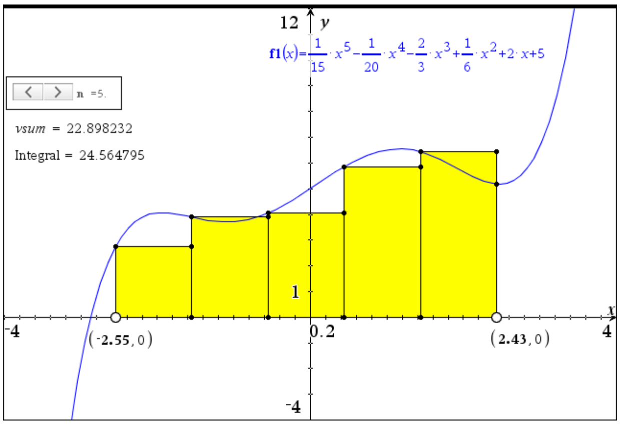
Til sidst trækkes rektanglerne op (som polygoner) og de farves gule.



Tilbage er der så blot at skjule punktplottene og tilføje værdien af venstresummen henholdsvis integralet. Venstresummen udregnes fx i regnearket som en celleformel:

$$D1 \quad \text{vsum}:=\text{sum}(\text{left}(\text{y\_int},\text{'n}))\cdot\frac{\text{'b}-\text{'a}}{\text{'n}}$$

Integralet udregnes nemt grafisk.



Har man overskud til det kan man nu tilføje højresummer, midtsummer osv. og udnytte **Betingelser** til at lade en skydervariabel vælge, hvilken type sum man vil have vist ☺

4 Regressionsmodeller

Regressionsmodeller omfatter de mest fundamentale vækstmodeller med funktioner: Lineær vækst, eksponentiel vækst og potensvækst. I **Lister og Regneark**-værkstedet har du adgang til et væld af værktøjer, der gør arbejdet med regressionsmodeller menustyret og meget fleksibelt. I dette afsnit vil du lære at

- plotte data i **Diagrammer og statistik**
- udføre lineær regression
- udføre eksponentiel regression
- udføre potens regression
- opstille andre former for regressionsmodeller

Lineær regression

En typisk eksamensopgave (omformuleret):
Skemaet viser trykket i forskellige dybder under havoverfladen

Dybde (m)	10	13	35	40	100
Tryk (atm)	1,96	2,25	4,36	4,84	10,6

Det oplyses, at trykket y med tilnærmelse er en lineær funktion $y = ax + b$ af dybden x.
Bestem tallene a og b.
Find trykket i en dybde på 150 m, og bestem den dybde, hvor trykket er 30 atm.

Data indtastes i et **Lister og Regneark**-værksted. Søjlerne navngives **dybde** og **tryk**.

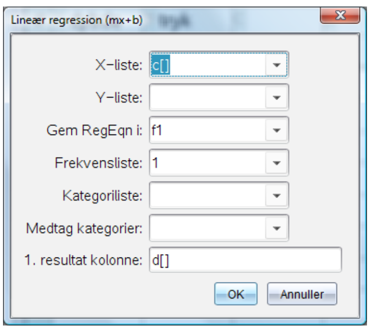
	A dybde	B tryk
=		
1	10	1.96
2	13	2.25
3	35	4.36
4	40	4.84
5	100	10.6

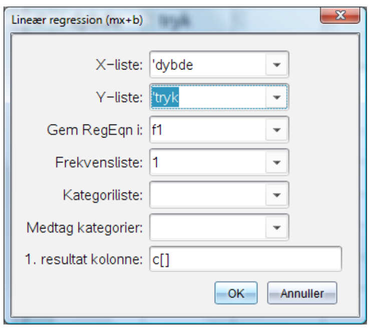
For at få udført lineær regression på disse data skal du vælge

4:Statistik ▸ 1:Statistiske beregninger ▸ 3:Lineær regression (mx+b)

Dette åbner en dialogboks, hvor der vælges **dybde** som **X-liste** og **tryk** som **Y-liste**:

Obs
Internt i **Lister og regneark** markeres variable med en apostrof ' foran navnet for at undgå forveksling med søjle- og cellenavne i regnearket.





I det tredje felt skal du angive det navn, som regressionsfunktionen skal have. Her foreslår programmet navnet **f1**. Uanset om man selv vælger et navn eller accepterer det foreslåede navn, bliver regressionsfunktionen tilgængelig i alle andre værksteder.

I det nederste felt **1. resultat kolonne** skrives her ***c*[ ]**. Det betyder, at resultatet af regressionen indsættes i regnearkets c- (og d-) søjle.

Efter klik på OK fremkommer resultatet:

	A dybde	B tryk	C	D
=				=LinRegMx('dybde,'tryk,1):
1	10	1.96	Titel	Lineær regression (mx+b)
2	13	2.25	RegEqn	m*x+b
3	35	4.36	m	0.09599
4	40	4.84	b	1.0008
5	100	10.6	r²	1.
6			r	1.
7			Resid	{⁻6.9976395342ᴇ⁻4,0.0013...

Koefficienterne i den lineære model $y = ax + b$ er altså $a = 0,09599$ og $b = 1,0008$.

For at besvare de næste to spørgsmål oprettes en side med **Noter**:

Tip
I en matematikboks kan man konvertere reduktionstegnet ▸ til et lighedstegn = under attributter.

Trykket i en dybde på 150 meter findes ved at indsætte 150 i funktionen f1

$\boxed{f1(150) = 15.3993}$

Altså er trykket 15,4 atm i en dybde på 150 m.

For at finde den dybde, hvor trykket er 30 atm, løses ligningen f1(x)=30:

solve(f1(x)=30,x) ▸ $\boxed{x=302.107}$

Altså skal man ned i en dybde på 302 m for at finde et tryk på 30 atm.

Resultatet af regressionen viser tillige to andre størrelser:

korrelationskoefficienten r og forklaringsgraden r^2.

I dette tilfælde er forklaringsgraden meget tæt på 1 eller 100 %. Det hænger sammen med, at data passer meget fint med antagelsen om en lineær model. Vi kommenterer betydningen af forklaringsgraden i et senere afsnit.

Grafisk modelkontrol

Hvis der ikke er givne oplysninger om modellen, skal valget af model begrundes. Det gøres meget overbevisende med en grafisk modelkontrol. Der oprettes et **Diagrammer og Statistik**-værksted, hvor **dybde** og **tryk** indsættes på akserne.

Tip
Ligningen for regressionslinjen i **Diagrammer og Statistik** vises kun når regressionslinjen er aktiv. For at vise den permanent skal den indskrives i en tekstboks.

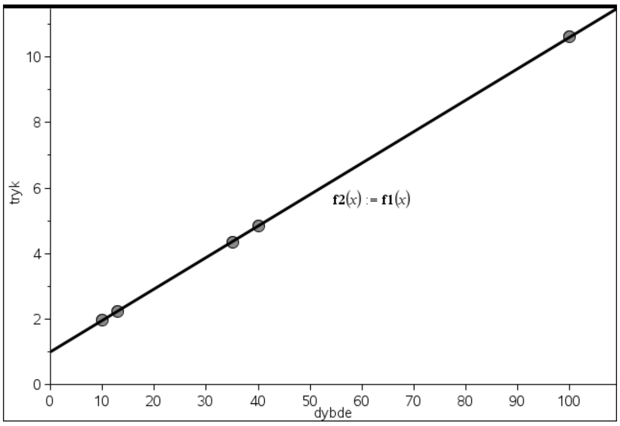
Regressionslinjen kan tegnes ved at plotte funktionen **f1**. Her skal du vælge **Plot funktion** fra **Undersøg data**-menuen (venstre skærmbillede). Man kan også tegne regressionslinjen ved at vælge menuen

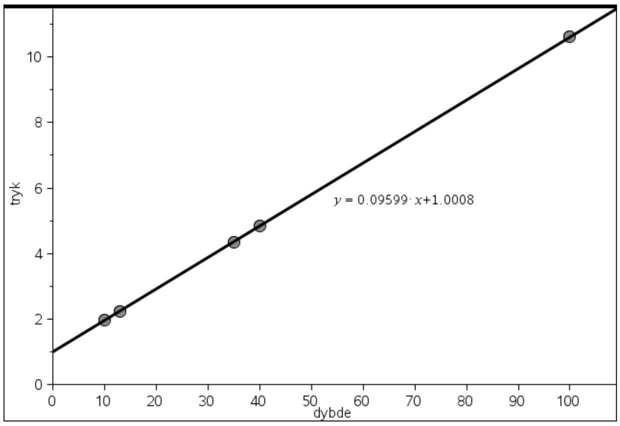
4:Undersøg data ▸ 6:Regression ▸ 1:Lineær regression (mx+b)

i **Diagrammer og Statistik**-vinduet (højre skærmbillede).

Vi ser, at datapunkterne ligger pænt på den rette linje. Den lineære model må derfor anses for at være en rimelig model.

Vi kan yderligere underbygge modellen ved at kigge på *residualplottet*, der inddrager de finere detaljer i datapunkternes fordeling omkring den rette linje.

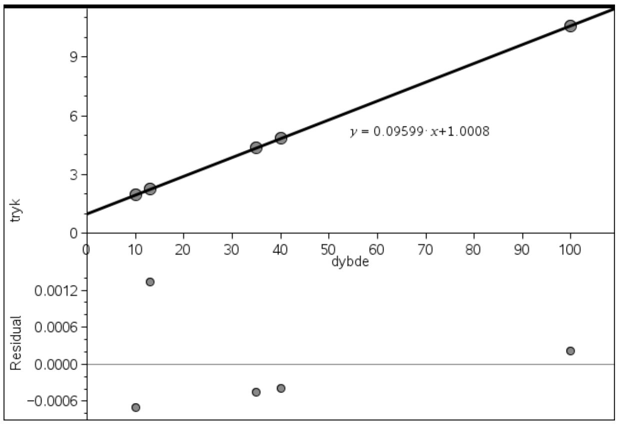




Residualplot

For at vise residualerne kan du arbejde videre med det foregående diagram.
Vælg

4: Undersøg data ▸ 7: Residualer ▸ 2: Vis residualplot



Obs
Residualerne er *afbalancerede*. Fx er summen af residualerne netop nul.

Residualplottet viser, at datapunkterne ligger rimelig tilfældigt fordelt omkring regressionslinjen, og afvigelserne mellem de målte værdier af trykket og modellens værdier er af størrelsesorden 0,001 atm og derunder. Med tilfældige afvigelser på under 1 promille må den lineære model siges at give en god beskrivelse af de foreliggende data.

Tegningen af regressionslinjen og det medfølgende residualplot udgør de vigtigste ingredienser i en grafisk modelkontrol.

Mindste kvadraters metode

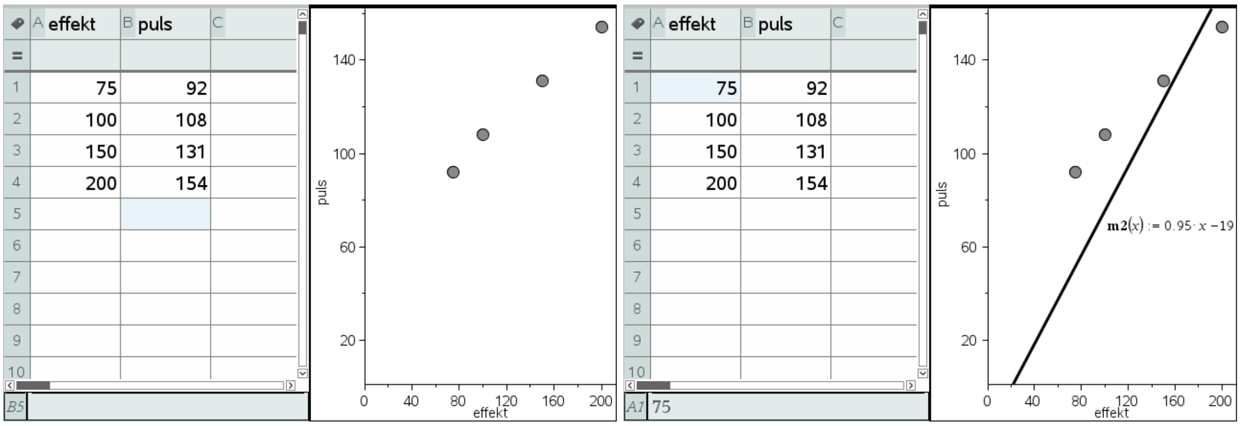
Vi kan også illustrere ideen bag den lineære regression. Vi må da skifte det ovenstående datasæt ud med et, som ikke er helt lige så overbevisende, for ellers er det svært at illustrere ideerne ☺ Vi ser derfor på det følgende datasæt:

En elev gennemfører en test på en ergometercykel (kondicykel).
Skemaet viser sammenhørende værdier af den effekt eleven yder og elevens puls

Effekt/Watt	75	100	150	200
Puls	92	108	131	154

Undersøg sammenhængen mellem pulsen og den ydede effekt.

Afsættes disse data i et **Lister og regneark**-værksted med tilhørende punktplot ses en rimelig, men *ikke* meget præcis lineær sammenhæng.

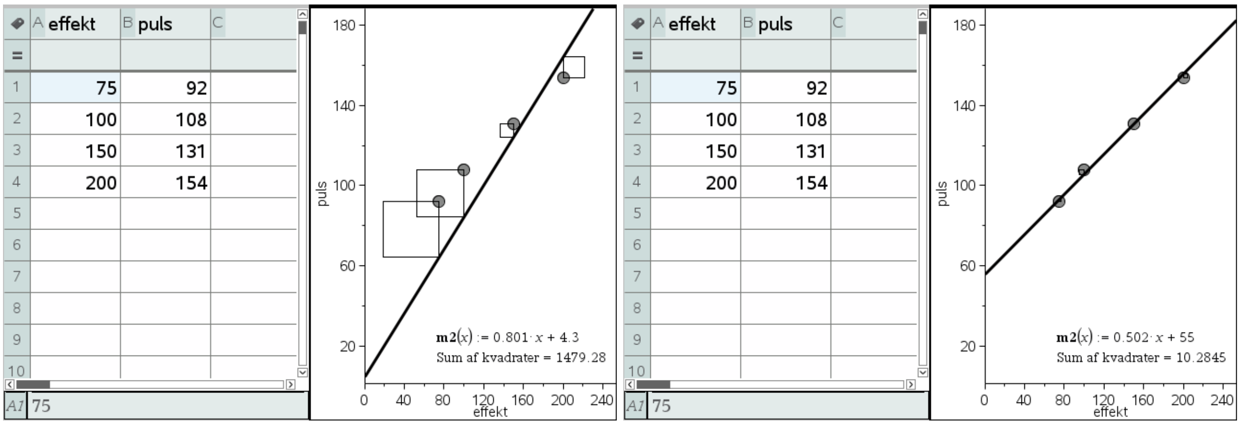


Vi kan nu som vist tilføje en flytbar ret linje til modellen ved at vælge

4: Undersøg data ▸ 2: Tilføj flytbare linjer

Denne linje kan flyttes rundt ved at gribe ude i enderne, hvor linjen drejes (dvs. man regulerer på hældningen), eller inde på midten, hvor linjen flyttes op og ned (dvs. man regulerer på startværdien/skæringen med anden-aksen). Samtidigt kan man tilføje residualkvadraterne ved at vælge

4: Undersøg data ▸ 7: Residualer ▸ 1: Vis residuelle kvadrater

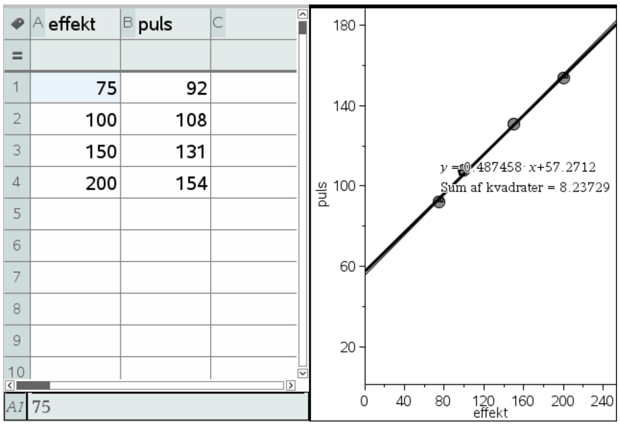


Vi ser da ikke bare kvadrater, der forbinder datapunkterne med den rette linje via den lodrette forbindelseslinje fra datapunktet til den rette linje, men også summen af disse kvadraters arealer, i dette tilfælde 1479.28. Ved at gøre *kvadraterne så små som muligt* (målt på deres samlede areal), kan man nu flytte linjen så tæt som muligt på datapunkterne. Her har vi fx reduceret kvadratsummen til 10.2845.

Denne metode til at finde den bedste rette linje gennem datapunkterne kaldes derfor også for mindste kvadraters metode.

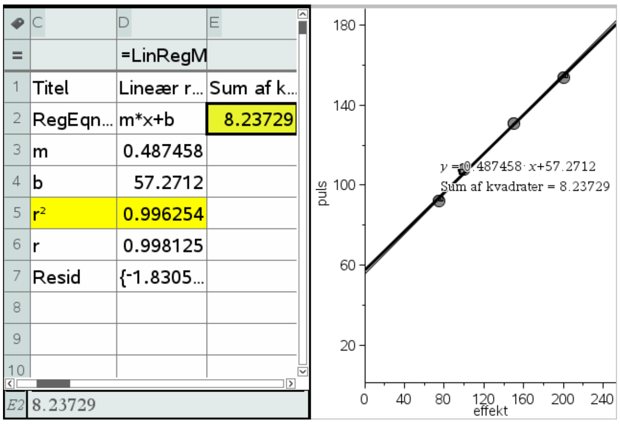
Vi kan betone sammenhængen med den lineære regressionsmodel ved at tilføje denne og efterfølgende tilføje de residuelle kvadrater til regressionslinjen. Vi finder da en kvadratsum som er endnu mindre, her 8.23729. Ligegyldigt hvordan vi placerer den flytbare rette linje vil vi nu opdage, at det ikke er muligt at komme under kvadratsummen for regressionslinjen: Regressionslinjen er nemlig konstrueret, så den har den mindst mulige kvadratsum ☺ Vi ser med andre ord

Regressionslinjen er fundet ved hjælp af mindste kvadraters metode.



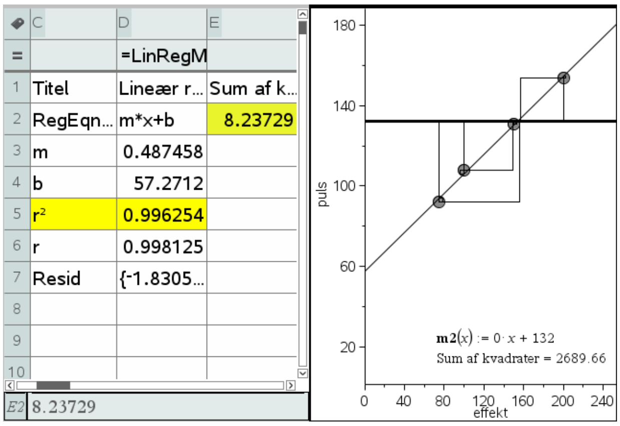
Forklaringsgraden

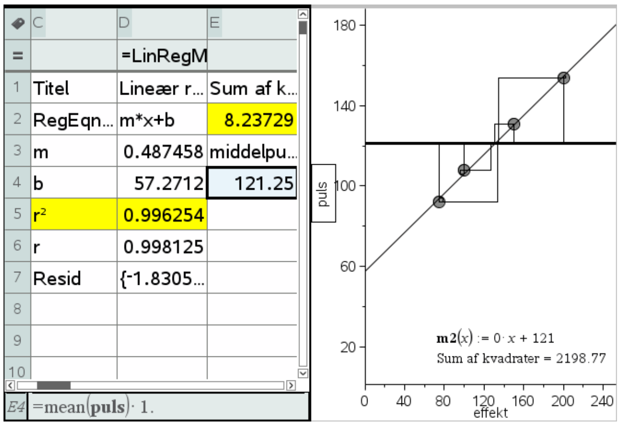
Vi har nu givet en forklaring på fremkomsten af den lineære regressionsmodel, men når vi udfører den lineære regression i regnearket fremkommer der yderligere oplysninger, herunder forklaringsgraden r^2, der i dette tilfælde er på 99.62%.



Vi vil nu prøve at forklare hvor dette tal kommer fra, og herunder prøve at kaste lys over betydningen af dette tal. Regressionslinjen fremkommer ved hjælp af mindste kvadraters metode, og i en vis forstand er der derfor tale om den bedste rette linje gennem datapunkterne. Men det siger jo ikke i sig selv noget om hvor god modellen er. Det kunne jo være, at *der slet ikke var nogen lineær sammenhæng mellem effekt og puls*, eller endnu værre: *At der slet ikke var nogen som helst sammenhæng mellem effekt og impuls*. Den sidste hypotese kaldes for *nulhypotesen* (jfr. hypotesetest, men her bruger vi altså begrebet indenfor rammerne af den deskriptive statistik). Hvis den lineære model giver en god beskrivelse af data, er det klart, at den må være væsentligt bedre end nulhypotesen, ellers kunne vi lige så godt bruge den til at beskrive data og konkludere, at den tilsyneladende lineære sammenhæng netop kun er tilsyneladende ☺

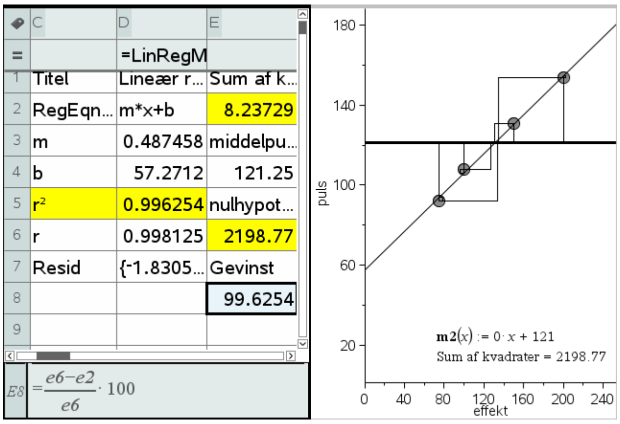
Hvis der slet ikke er nogen sammenhæng mellem effekt og puls (nulhypotesen), dvs. pulsen svinger bare tilfældigt op og ned uden indflydelse fra effekten, så er den funktion, der beskriver sammenhængen (eller rettere den manglende sammenhæng) *konstant*, dvs. afhænger ikke af effekten. Vi griber derfor den flytbare rette linje og retter den op, til den er vandret. Det kræver lidt behændighed, men det er altså afgørende for at den flytbare rette linje nu kan repræsentere nulhypotesen. Vi kan da stadigvæk regulere på højden af den vandrette linje og justerer højden indtil den tilhørende kvadratsum er så lille som muligt, her 2198.77 ved en højde på 121. De 121 svarer da netop til middelværdien af pulsen, da vi skal ligge så tæt på pulserne som muligt. Udregnes middelpulsen fås som vist 121.25, men da det er en grafisk metode kan vi ikke ramme middelværdien tættere end 1 pixels afstand ☺





Vi ser da at kvadratsummen hørende til den bedste vandrette linje er 2198.77. Men kvadratsummen hørende til bedste skrå rette linje var kun 8.23729, dvs. væsentligt mindre. Det er jo netop, hvad vi må forvente, hvis vi skal tro på en lineær sammenhæng. Hvis de var af samme størrelsesorden kunne vi jo lige så godt antage, at der slet ikke var nogen sammenhæng.

Vi kan også indføre et mål for hvor meget bedre *den bedste skrå rette linje* (regressionsmodellen) er sammenlignet med *den bedste vandrette rette linje* (nulhypotesen). Sætter vi kvadratsummen for nulhypotesen til 100%, så kan vi udregne hvor mange procent, vi fjerner, ved at overgå til regressionsmodellen:



Men dermed finder vi jo netop forklaringsgraden r^2, der altså kan ses som et mål for hvor god regressionslinjen er i forhold til nulhypotesen (at der slet ikke er nogen sammenhæng).

Summen af de residuelle kvadrater kaldes ofte for *variationen*. Vi kan altså fjerne 99.6254% af variationen i datasættet ved at gå fra ingen sammenhæng til en lineær sammenhæng.

Hvad så med korrelationskoefficienten r? Der skal vi bare bemærke, at forklaringsgraden netop er kvadratet på korrelationskoefficienten. Det er altså kun fortegnet for korrelationskoefficienten vi mangler at gøre rede for. Men det er nemt nok: Hvis hældningen for regressionlinjen er positiv, er korrelationen postiv, og hvis hældningen af regressionslinjen er negativ, er korrelationen negativ.

Men i modsætning til korrelationen, kan forklaringsgraden anvendes på en vilkårlig model. Hvis vi beskriver datasættet ved hjælp af en eller anden modelfunktion $y = f(x)$ kan vi nemlig altid udregne summen af de residuelle kvadrater og sammenholde dem med summen af de residuelle kvadrater for den bedste vandrette linje.

Eksponentiel regression

En typisk eksamensopgave (omformuleret):
Tabellen viser sammenhørende værdier af temperaturen T (målt i °C) i en fryser og holdbarheden D (målt i dage) af en rullepølse, der opbevares i en fryser.

Temperatur (°C)	–25	–20	–15	–10
Holdbarhed (døgn)	280	154	91	49

Det oplyses, at D med god tilnærmelse er en eksponentielt aftagende funktion af T.

a) Bestem en forskrift for denne funktion.

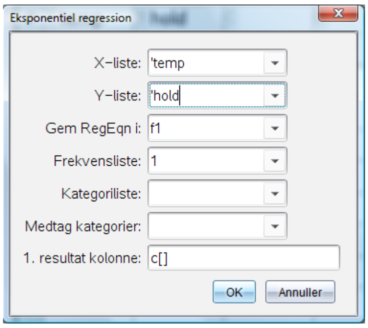
b) Bestem ved hjælp af denne forskrift holdbarheden ved en temperatur på –18°C, og bestem temperaturen, hvis holdbarheden er 180 døgn.

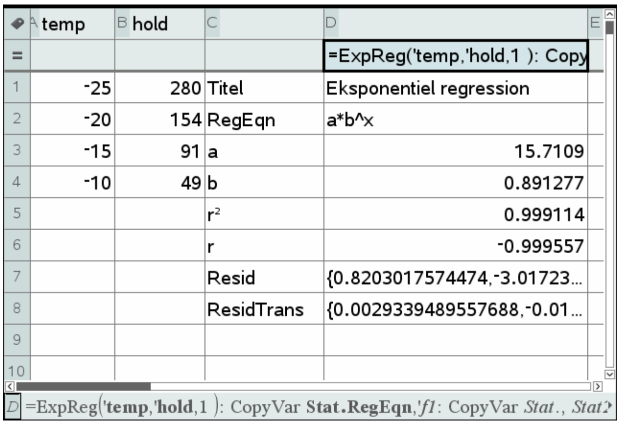
c) Bestem ved hjælp af den fundne forskrift halveringstiden for holdbarheden, og bestem den procentvise ændring i holdbarheden, når temperaturen øges 2 °C.

Tast data ind i et **Lister og Regneark**-værksted, navngiv søjlerne, og udfør regressionen i regnearket med

Tip
Brug beskrivende navne til listerne. Undgå at bruge navne med ét bogstav, da de kan forveksles med søjlenavnene i regnearket.

$\bar{X}$ **4:Statistik ▸ 1:Statistiske beregninger ▸ A:Eksponentiel regression…**





	A temp	B hold	C	D
=				=ExpReg('temp,'hold,1): Copy
1	-25	280	Titel	Eksponentiel regression
2	-20	154	RegEqn	a*b^x
3	-15	91	a	15.7109
4	-10	49	b	0.891277
5			r²	0.999114
6			r	-0.999557
7			Resid	{0.8203017574474,-3.01723…
8			ResidTrans	{0.0029339489557688,-0.01…
9				
10				

D =ExpReg('temp,'hold,1): CopyVar Stat.RegEqn,'f1: CopyVar Stat., Stat.

Obs
TI-Nspire CAS giver forskriften på formen $a \cdot b^x$, dvs. tallene a og b er byttet om i forhold til normal dansk notation.

Resten af besvarelsen skrives i **Noter**. Tallene a og b i forskriften bliver automatisk gemt under navnene **stat.a** og **stat.b**. Det kan undertiden betale sig at gemme disse værdier med de sædvanlige danske betegnelser, b og a jf. c) nedenfor.

Tip
De variable, der er gemt, kan findes i variabelregistret ved at trykke på knappen

var

Man kan også taste **stat.** i et matematikfelt. Så dukker der en liste op over de gemte statistiske variable i opgaven.

a) Ved eksponentiel regression har vi fundet forskriften: $\mathbf{f1}(x) = 15.7109 \cdot (0.891277)^x$.

Holdbarheden som funktion af temperaturen er derfor givet ved

$D=15.7109 \cdot (0.891277)^T$

b) For at finde holdbarheden ved −18°C beregnes $\mathbf{f1}(-18) = 124.731$.

Altså er holdbarheden ca. 125 døgn, når rullepølsen opbevares ved −18°C.

For at finde den temperatur, der skal til, for at rullepølsen kan have en holdbarhed på 180 døgn, løses ligningen f1(x)=180:

$\text{solve}(\mathbf{f1}(t)=180,t) \blacktriangleright t=-21.1868$

En holdbarhed på 180 døgn kræver altså en temperatur på ca. −21°C.

c) Funktionens parametre gemmes under de sædvanlige navne:

$a:=\mathbf{stat.b} \blacktriangleright 0.891277 \quad b:=\mathbf{stat.a} \blacktriangleright 15.7109,$

dvs. $\mathbf{a} = 0.891277$ og $\mathbf{b} = 15.7109$.

Halveringskonstanten $T_{1/2}$ bestemmes ved formlen $T_{1/2} = \dfrac{\ln(0.5)}{\ln(\mathbf{a})}$:

$\dfrac{\ln(0.5)}{\ln(\mathbf{a})} = 6.02213$

Halveringskonstanten er altså ca. 6°C. Det betyder, at holdbarheden halveres, for hver gang opbevaringstemperaturen øges med 6°C.

Når temperaturen øges med 2°C, ændres holdbareheden med $(\mathbf{a}^2-1) \cdot 100\%$:

$(\mathbf{a}^2-1) \cdot 100 = -20.5625$

Holdbarheden falder altså med ca. 20,6%, hver gang temperaturen øges med 2°C.

Metoden bag eksponentiel regression: Logaritmerne kommer på banen

Spørgsmålet er så hvordan TI-Nspire CAS udfører den eksponentielle regression. Man kunne tro programmet brugte mindste kvadraters metode direkte, men det gør programmet ikke. Ved regressionsmodeller er der tradition for at man i stedet om muligt *transformerer data*, så de i stedet følger *en lineær sammenhæng*. Derefter udfører man en sædvanlig lineær regression på de transformerede data. Til sidst transformerer man så den lineære ligning tilbage til de oprindelige data.

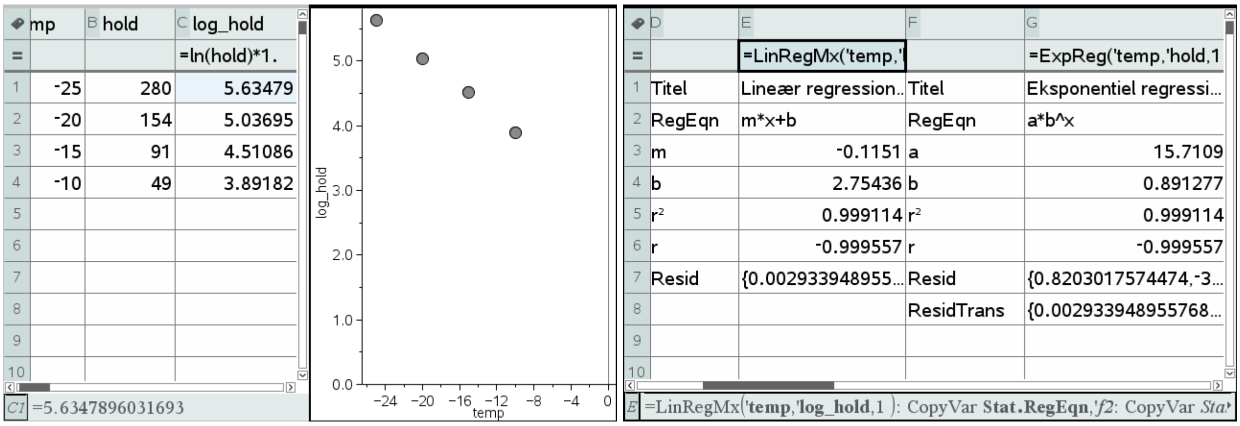
Vi kan se i regnearket, at det er denne strategi, der er fulgt, for der oplyses to residualer **Resid** og **ResidTrans**. Her står **Resid** for residualerne for de oprindelige data, mens **ResidTrans** står for residualerne for de transformerede lineære data.

Tilbage er så blot spørgsmålet om hvordan man transformerer en eksponentiel vækst $y = b \cdot a^x$ om til en lineær vækst? Det er her logaritmerne kommer på banen ☺ Ifølge logaritmeregnereglerne gælder der nemlig

$$\begin{aligned}\ln(y) &= \ln(b \cdot a^x) = \ln(b) + x \cdot \ln(a)\\ \ln(y) &= \ln(a) \cdot x + \ln(b)\\ Y &= \ln(a) \cdot x + \ln(b)\end{aligned}$$

Hvis vi skifter til de transformerede data $(x,Y) = (x, \ln(y))$ er der derfor netop tale om en lineær sammenhæng.

Vi skal altså finde logaritmen til holdbarheden og indfører derfor en ny søjle **log_hold** for de transformerede data



Som det ses synes de transformerede punkter netop at ligge på ret linje, hvorfor vi som vist udfører en lineær regression på de transformerede data. Vi ser da, at forklarings-graden r^2 og korrelationskoefficienten r netop er fælles for den lineære regression og den eksponentielle regression, der altså *arver* disse værdier fra den lineære regression. Vi ser også, at residualerne for den lineære regression netop svarer til de transformere-de residualer i den eksponentielle regression.

Tilbage står så blot at transformere ligningen for den lineære regression, dvs. f2, tilbage til de oprindelige data. Det gøres nemmest i et **Note**-værksted:

Først opskrives den transformerede ligning:

$\ln(y)=\mathbf{f2}(x) \blacktriangleright \ln(y)=2.75436-0.1151\cdot x$

Det er nærliggende at solve for at isolere y – men for en gangs skyld går det galt!

$\text{solve}(\ln(y)=\mathbf{f2}(x),y) \blacktriangleright y=(1.)^{5.755\text{E}12\cdot x-1.37718\text{E}14}$

Vi transformerer derfor ligningen tilbage ved at bruge den omvendte logaritme, dvs. den naturlige eksponentialfunktion

$y=\mathbf{e}^{\mathbf{f2}(x)} \blacktriangleright y=15.7109\cdot(0.891277)^x$

Sammenholder vi det med resultatet fra den eksponetielle regression, dvs. y = f1(x), ser vi netop at der er fuld overensstemmelse

$\mathbf{f1}(x) \blacktriangleright 15.7109\cdot(0.891277)^x$

Så nu ved vi hvordan den eksponentielle regression virker ☺

Potensregression

En typisk eksamensopgave (omformuleret):
Tabellen viser for 3-slået tovværk sammenhængen mellem tovværkets diameter (målt i mm) og tovets brudstyrke (målt i kg).

Diameter	4	5	6	8	10	14	16	20	24	26
Brudstyrke	250	400	600	1000	1550	3200	4000	6000	8600	10 000

Det oplyses, at brudstyrken som funktion af diameteren kan beskrives ved en model af formen $f(x)=b\cdot x^a$.

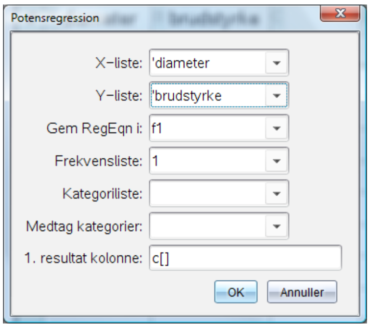
a) Bestem tallene a og b.

b) Hvor mange procent skal diameteren ifølge denne model øges, hvis brudstyrken skal fordobles?

4:Regressionsmodeller

Tast data ind i et **Lister og Regneark**-værksted, navngiv søjlerne, og udfør regressionen i regnearket med

4:Statistik ▸ 1:Statistiske beregninger ▸ 9:Potensregression…



Obs
TI-Nspire CAS giver forskriften på formen $a \cdot x^b$, dvs. tallene a og b er byttet om i forhold til normal dansk notation.

	A diameter	B brudstyrke	C	D	E
=				=PowerReg('diamet	
1	4	250	Titel	Potensregression	
2	5	400	RegEqn	a*x^b	
3	6	600	a	17.0922	
4	8	1000	b	1.96192	
5	10	1550	r²	0.999511	
6	14	3200	r	0.999756	
7	16	4000	Resid	{-9.413487917570…	
8	20	6000	ResidTrans	{-0.036962349266…	
9	24	8600			
10	26	10000			

D =PowerReg('diameter,'brudstyrke,1): CopyVar Stat.RegEqn,'f1: CopyV▸

Resten af besvarelsen skrives i **Noter**. Se tip på den foregående side om statistiske variable.

a) Potensregressionen ovenfor viser, at modellen er givet ved

$f(x) = 17.0922 \cdot x^{1.96192}$

Funktionens parametre gemmes under de sædvanlige navne

a:=stat.b ▸ 1.96192 og b:=stat.a ▸ 17.0922,

dvs. resultatet bliver a = 1.96192 og b = 17.0922

b) Når diameteren ganges med faktoren k bliver brusdstyrken k^a gange så stor.

Vi ønsker at faktoren k^a skal være 2, dvs. vi skal løse ligningen $k^a = 2$:

solve$(k^a=2,k)$ ▸ $k=1.42376$

Heraf ses, at

Diameteren skal øges med 42%, hvis brudstyrken skal fordobles.

Metoden bag potensregression: Logaritmerne kommer på banen

Spørgsmålet er så hvordan TI-Nspire CAS udfører potensregressionen. Igen kunne man tro, at programmet brugte mindste kvadraters metode direkte, men det gør programmet ikke. Ved regressionsmodeller er der tradition for at man i stedet om muligt *transformerer data*, så de i stedet følger *en lineær sammenhæng*. Derefter udfører man en sædvanlig lineær regression på de transformerede data. Til sidst transformerer man så den lineære ligning tilbage til de oprindelige data.

Vi kan netop se i regnearket, at det er denne strategi, der er fulgt, for der oplyses to residualer **Resid** og **ResidTrans**. Her står **Resid** for residualerne for de oprindelige data, mens **ResidTrans** står for residualerne for de transformerede lineære data.

Tilbage er så blot spørgsmålet om hvordan man transformerer en potensvækst $y = b \cdot x^a$ om til en lineær vækst? Igen er det logaritmerne, der kommer på banen ☺ Ifølge logaritmeregnereglerne gælder der nemlig

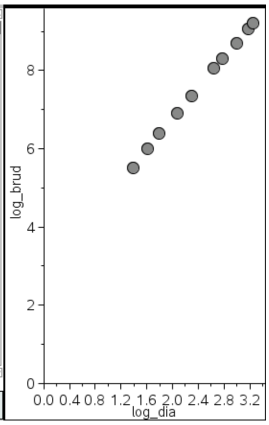
$$\ln(y) = \ln(b \cdot x^a) = \ln(b) + a \cdot \ln(x)$$
$$\ln(y) = a \cdot \ln(x) + \ln(b)$$
$$Y = a \cdot X + \ln(b)$$

Hvis vi skifter til de transformerede data $(x, Y) = (\ln(x), \ln(y))$ er der derfor netop tale om en lineær sammenhæng. Læg mærke til, at hældningen for denne lineære sammenhæng netop er potensen i potenssammenhængen.

Vi skal altså finde logaritmerne til såvel diameteren som brudstyrken og indfører derfor to ny søjler **log_dia** og **log_brud** for de transformerede data

	A dia...	B brud...	C log_dia	D log_brud
=			=ln(diame	=ln(brudsty
1	4	250	1.38629	5.52146
2	5	400	1.60944	5.99146
3	6	600	1.79176	6.39693
4	8	1000	2.07944	6.90776
5	10	1550	2.30259	7.34601
6	14	3200	2.63906	8.07091
7	16	4000	2.77259	8.29405
8	20	6000	2.99573	8.69951
9	24	8600	3.17805	9.05952
10	26	10000	3.2581	9.21034

D log_brud:=ln(brudstyrke)· 1.



	E	F	G	H
=		=LinRegMx('log_dia,'l		=PowerReg('diamet
1	Titel	Lineær regression (...	Titel	Potensregression
2	RegEqn	m*x+b	RegEqn	a*x^b
3	m	1.96192	a	17.0922
4	b	2.83862	b	1.96192
5	r^2	0.999511	r^2	0.999511
6	r	0.999756	r	0.999756
7	Resid	{-0.0369623492664...	Resid	{-9.413487917570...
8			ResidTrans	{-0.036962349266...
9				
10				

F =LinRegMx('log_dia,'log_brud,1): CopyVar Stat.RegEqn,'f2: CopyVar

Som det ses synes de transformerede punkter netop at ligge på ret linje, hvorfor vi som vist udfører en lineær regression på de transformerede data. Vi ser da, at forklaringsgraden r^2 og korrelationskoefficienten r netop er fælles for den lineære regression og den eksponentielle regression, der altså *arver* disse værdier fra den lineære regression. Vi ser også, at residualerne for den lineære regression netop svarer til de transformerede residualer i potensregressionen.

Vi lægger også mærke til, at hældningen for den lineære regressionsmodel (1.96192) netop svarer til potensen for potensregressionen.

Tilbage står så blot at transformere ligningen for den lineære regression, dvs. f2, tilbage til de oprindelige data. Det gøres nemmest i et **Note**-værksted:

Først opskrives den transformerede ligning:

$\ln(y)=\mathbf{f2}(\ln(x))$ ▸ $\ln(y)=1.96192\cdot \ln(x)+2.83862$

Det er nærliggende at solve for at isolere *y* – og denne gang går det godt!

$\text{solve}(\ln(y)=\mathbf{f2}(\ln(x)),y)$ ▸ $y=17.0922\cdot x^{1.96192}$ and $x\geq 0.$

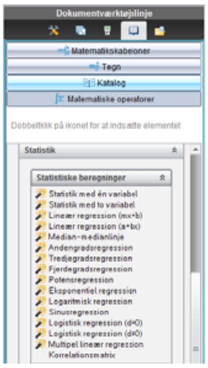
Men man kan også transformere ligningen tilbage ved hjælp af den naturlige eksponentialfunktion:

$y=e^{\mathbf{f2}(\ln(x))}$ ▸ $y=17.0922\cdot x^{1.96192}$ ⚠

Sammenholder vi det med resultatet fra potensregressionen, dvs. *y* = f1(*x*), ser vi netop, at der er fuld overensstemmelse

$\mathbf{f1}(x)$ ▸ $17.0922\cdot x^{1.96192}$

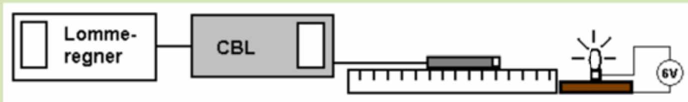
*Andre regressionsmodeller**



Hidtil har vi kun set på de tre klassiske regressionsmodeller: Den lineære regression, den eksponentielle regression og potensregressionen. Men der findes mange andre indbygget i TI-Nspire CAS, fx polynomiale regressionsmodeller (andengradspolynomier, tredjegradspolynomier og fjerdegradspolynomier), trigonometriske regressionsmodeller og logistiske regressionsmodeller. De bygger alle på mindste kvadraters metode.

Men der er også mange standardmodeller, som *ikke* er indbygget i TI-Nspire CAS. Der må man altså sno sig ☺ En mulighed er, at finde en snedig variabeltransformation, som transformerer den søgte sammenhæng over i en lineær sammenhæng. En anden mulighed er, at prøve at udvide den lineære regressionsmodel med endnu en parameter, som fastlægges grafisk. Vi viser eksempler på begge metoder:

Forskudt omvendt kvadratisk proportionalitet*



Da det areal lyset spredes over vokser med kvadratet på afstanden til lyskilden, forventer vi at lysintensiteten er omvendt proportional med kvadratet på afstanden.
I et konkret forsøg fik man følgende data

Afstand (cm)	10	15	20	25	30	35	40
Lysintensitet (mW/cm^2)	0.631	0.399	0.255	0.178	0.134	0.107	0.095

Undersøg om lysintensiteten er omvendt proportional med kvadratet på afstanden.

I virkeligheden ligger lysmåleren et stykke inde i sonden, og pæren står et stykke inde på brættet, hvorfor vi skal lægge en overskydende længde *d* til afstanden til pæren.

Vi skal derfor modellere datasættet med en sammenhæng af formen:

$$\text{Lysintensitet} = \frac{k}{(\text{afstand} + d)^2}$$

$$\text{afstand} = \frac{\sqrt{k}}{\sqrt{\text{Lysintensitet}}} - d$$

Vi ser derfor at afstanden er en lineær funktion af den reciprokke kvadratrod af lysintensiteten. Sætter vi med andre ord $y = \text{afstand}$ og $x = \dfrac{1}{\sqrt{\text{lysintensitet}}}$ fås en lineær sammenhæng på formen

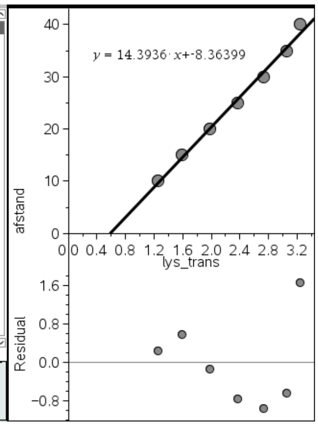
$$y = a \cdot x + b \quad \text{, hvor } a = \sqrt{k} \text{ og } b = -d$$

Her er hældningen knap så interessant, mens skæringen med andenaksen viser hvor stor en nulpunktsfejl, der er på målestokken.

Vi skriver derfor data ind i et **Lister og regneark**-værksted og transformerer data, afbilder de transformerede data i et **Diagram og statistik**-vindue for tilsidst at udføre en lineær regression på de transformerede data

	A afstand	B lysinte...	C lys_tra...
=			=1/(sqrt('ly
1	10	0.631	1.25888
2	15	0.399	1.58312
3	20	0.255	1.9803
4	25	0.178	2.37023
5	30	0.134	2.73179
6	35	0.107	3.05709
7	40	0.095	3.24443
8			
9			

C $lys\_trans := \frac{1}{\sqrt{'lysintensitet}}$



Derefter kan vi transformere ligningen tilbage til de oprindelige data, hvilket gøres nemmest i et **Note**-værksted.

stat.results ▸
$$\begin{bmatrix} \text{"Titel"} & \text{"Lineær regression (mx+b)"} \\ \text{"RegEqn"} & \text{"m·x+b"} \\ \text{"m"} & 14.3936 \\ \text{"b"} & -8.36399 \\ \text{"r²"} & 0.992754 \\ \text{"r"} & 0.99637 \\ \text{"Resid"} & \text{"{...}"} \end{bmatrix}$$

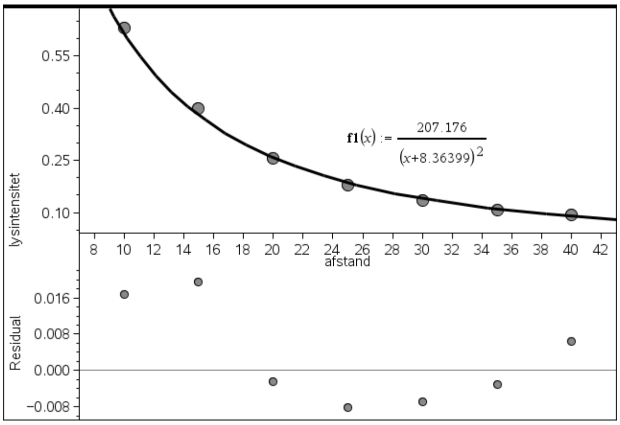
Da vi allerede har navngivet variablene afstand og lysintensitet i **Lister og regneark**–værkstedet sætter vi et understregningstegn _ bag variabelnavnene i **Noter**:

$$afstand\_=\mathbf{stat.RegEqn}\left(\frac{1}{\sqrt{lysintensitet\_}}\right) ▸ afstand\_=\frac{14.3936}{\sqrt{lysintensitet\_}}-8.36399$$

$$\text{solve}\left(afstand\_=\frac{14.3936}{\sqrt{lysintensitet\_}}-8.36399, lysintensitet\_\right) ▸ lysintensitet\_=\frac{207.176}{(afstand\_+8.36399)^2}$$

Når afstanden korrigeres for nulpunktsfejlen er der en overbevisende omvendt kvadratisk sammenhæng mellem afstand en fra lyskilden og lysintensiteten i denne afstand.

Forklaringsgraden r^2 er på 99.2754% og residualerne hører til de transformerede data. Men vi kan godt tegne grafen for den forskudte omvendte kvadratiske proportionalitet ind sammen med de oprindelige data, herunder få vist residualerne for de oprindelige data:



Der er altså rige muligheder for grafisk modelkontrol. Med typiske afvigelse på omkring 2% synes modellen absolut rimelig.

En transformation til lineære data er selvfølgelig kun mulig, hvis modellen kun indeholder to frie parametre, der efter transformationen kan indgå i hældningen og skæringen med andenaksen. Modeller med fx tre parametre kan derfor *ikke* håndteres på den ovenstående måde. Det gælder fx logistisk vækst og harmoniske svingninger, der i stedet håndteres iterativt.

Forskudt eksponentiel vækst*

Som et eksempel på en model med tre frie parametre ser vi til sidst på den forskudte eksponentielle vækst

$$y = b \cdot a^x + c$$

Denne model optræder fx i forbindelse med Newtons afkølingslov.

Ved afkøling af en kop te finder man fx de følgende data for teens temperatur som funktion af tiden:

Tid i minutter	0	15	30	45	60	90
Temperatur i °C	88.0	79.5	72.2	65.8	60.3	51.3

I følge Newtons afkølingslov nærmer teens temperatur sig stuetemperaturen eksponentielt, dvs. via en afhængighed af formen $y = b \cdot a^x + c$, hvor c repræsenterer stuetemperaturen.

Undersøg om de ovenstående data med rimelighed kan beskrives ved denne model.

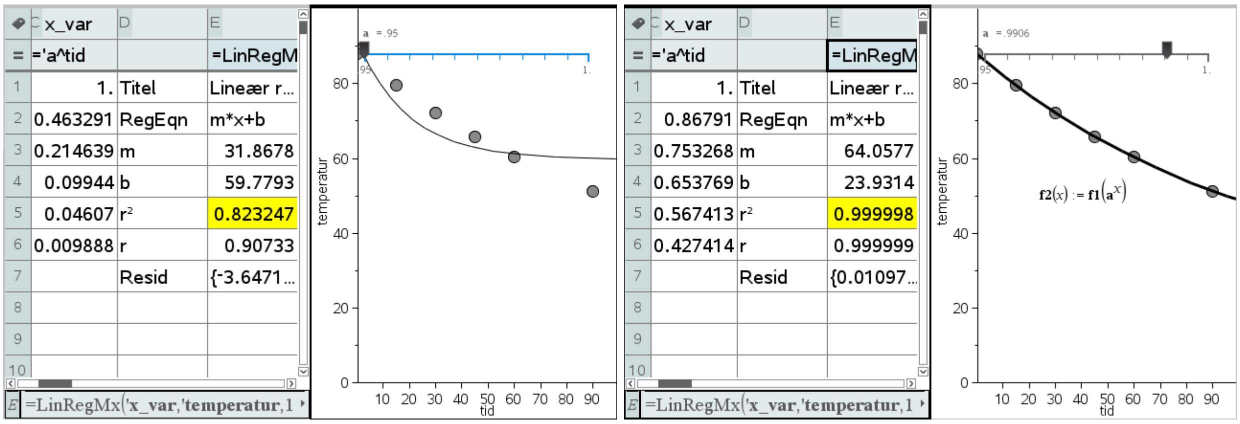
To af modellens parametre er lineære. Hvis vi omskriver modellen på formen

$$y = b \cdot X + c \quad \text{med} \quad X = a^x$$

kan vi føre den tilbage til en lineær model, hvor den sidste ikke-lineære parameter *a* er gemt i den transformerede variabel. Vi opretter derfor en skyder for *a* (der går fra 0.95 til 1 i trin af 0.0001), transformerer dataene som vist i formlen, dvs.

$$\mathbf{x\_var} = a^{\mathbf{tid}}$$

og afbilder de *oprindelige* data (**tid**, **temperatur**) grafisk, samtidigt med at vi udfører en lineær regression på de *transformerede* data $(\mathbf{x\_var}, \mathbf{temperatur})$, hvor regressionsfunktionen gemmes i **f1**:



Det gælder nu om at holde tungen lige i munden. For en given værdi af grundtallet *a* finder vi de bedste værdier for *b* og *c* med mindste kvadraters metode (den lineære regression). Samtidigt får vi oplyst forklaringsgraden for den lineære regression (gul celle). Jo større forklaringsgraden er, jo mindre er kvadratsummen. Ved at køre på skyderen for *a* kan vi derfor finde den værdi af *a*, der har den største forklaringsgrad og dermed den mindste kvadratsum. Vi finder altså modellen

$$\mathbf{temperatur} = 64.0577 \cdot 0.9906^{\mathbf{tid}} + 23.9314$$

Samtidigt ser vi at modellen har en forklaringsgrad r^2 på 99.9998% og grafen passer fortrinligt. Vi har altså gennemført en regressionsmodel baseret direkte på mindste kvadraters metode for en forskudt eksponentiel vækstmodel ☺.

5 Deskriptiv statistik

Iden deskriptive statistik arbejder man med **simple diskrete variable**, dvs. variable, der kun har *et endeligt antal værdier*. Der findes også **tællelige diskrete variable**, men dem kommer vi ikke nærmere ind på her. Diskrete variable står i modsætning til **kontinuerte variable**, der antager alle værdier *indenfor et interval*.

Kontinuerte variable undersøges ved hjælp af funktioner, differential- og integralregning, mens diskrete variable undersøges ved hjælp af lister og tabeller og summer og differenser.

Diskrete variable og datasæt

De simple diskrete variable kommer nu i to hovedtyper:

1) De **kategoriske variable**, hvis værdier er tekststrenge, der *altid* indskrives med gåseøjne!
2) De **numeriske variable**, hvis værdier er tal. Særligt simpelt er det, hvis værdierne er hele tal, men det er ikke noget krav.

Med de kategoriske variable kan man kun foretage optællinger og udregne simple procentdele. De numeriske variable kan man derimod regne på, herunder udregne vigtige *statistiske deskriptorer*, som eksempelvis kan angive *niveauet* ved hjælp af eksempelvis median og middelværdi, henholdsvis *variationen* ved hjælp af eksempelvis variationsbredde, kvartilbredde og spredning.

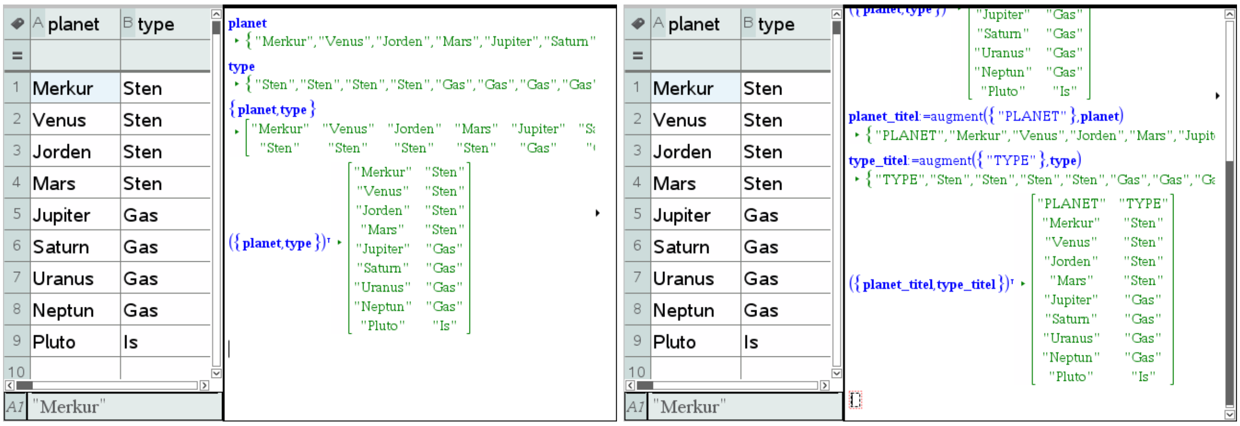
Der findes også en tredje type diskret variabel, der ligger midt mellem de kategoriske variable og de numeriske variable. De kaldes **ordinale variable**, og her drejer det sig typisk om kategoriske variable, der kan rangordnes, dvs. man kan sige hvilke værdier, der går forud for hvilke værdier. Det kan fx være resultatet af en spørgeskemaundersøgelse, hvor svarene rangordnes: Hvor enig er du i følgende udsagn …, som kan have værdierne ”meget uenig”, ”lidt uenig”, ”neutral”, ”lidt enig” og ”meget enig”. I mange tilfælde vil man da også kode værdierne som tal, eksempelvis -2, -1, 0, 1, 2, fordi man så også kan udføre formelle udregninger på værdierne, selvom man kan diskutere hvor stor mening det egentlig giver. Men det giver et hurtigt overblik over hvorvidt besvarelserne i én spørgeskemaundersøgelse er mere enige i udsagnet, end besvarelserne i en anden spørgeskemaundersøgelse.

TI-Nspire CAS understøtter dog *ikke* ordinale variable som en særskilt variabeltype.

Når man arbejder med flere variable hørende til det samme **datasæt** kan de med fordel opskrives som en **tabel** i et **Lister og regneark**-værksted. De enkelte variable afsættes da som lister med *lige mange elementer*. Mangler der nogle data, markeres det med et understregningstegn i en i øvrigt tom celle, dvs. _ . Et sådant datasæt kan også afbildes i fx et **Note**-værksted. Listerne kan da enten afbildes enkeltvis eller samlet, idet de omsluttes af krøllede parenteser, da en tabel/datasæt, jo består af **en liste af lister**.

Man skal da blot huske på, at man selv skal tilføje variabelnavnene inde i tabellen ved hjælp af sammenkæd-kommandoen augment, ligesom man skal huske på, at listerne nu som standard skrives vandret. Det sidste kan man dog rette op på med en transponeringskommando T (som du finder under **Tegn** i venstre sidepanel).

Her er det vist med et meget simpelt eksempel på et datasæt, der bygger på data om planeterne i solsystemet:



Diagrammer hørende til deskriptiv statistik

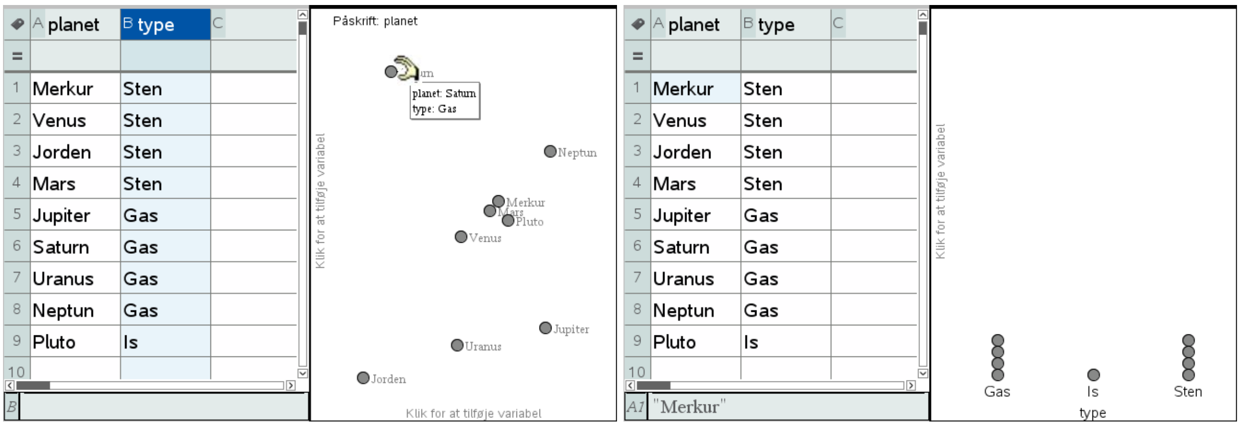
For at danne sig et overblik over hvordan værdierne for en diskret variabel fordeler sig, er det vigtigt at kunne visualisere deres fordeling i et passende diagram. TI-Nspire CAS understøtter en lang række diagrammer til dette formål. Vi starter med at danne os et overblik over hvilke diagramtyper, der er til rådighed.

Vi må da skelne mellem enkelt-variabelstatistik, dobbelt-variabelstatistik og multivariabel-statistik, der har hver deres egne karakteristiske diagrammer til rådighed.

Ved enkeltvariabelstatistik er det alene typen, der afgør, hvilke diagrammer, der er til rådighed: Drejer det sig om kategoriske eller numeriske data. Ved mere end én variabel må vi også skelne **uafhængige og afhængige variable**.

Ustrukturerede diagrammer

Åbner man et **Diagrammer og statistik**-værksted, vil det som standard åbne et *ustruktureret diagram*, hvor de data-lister der er til stede repræsenteres af tilfældigt placerede kugler med etiketter hørende til de enkelte datakugler.



Man kan skifte etiket ved at klikke i feltet **påskrift** øverst til venstre i diagramvinduet. Vi kan i dette tilfælde skifte til etiketten type. Man kan derefter strukturere diagrammet ved at trække en kugle ind på en af akserne. Trækker man fx en kugle ind på den lodrette akse stables kuglerne som vist lodret efter den valgte etiket.

Der findes selvfølgelig også mere systematiske metoder til at frembringe strukturerede diagrammer, som vi nu vil se nærmere på.

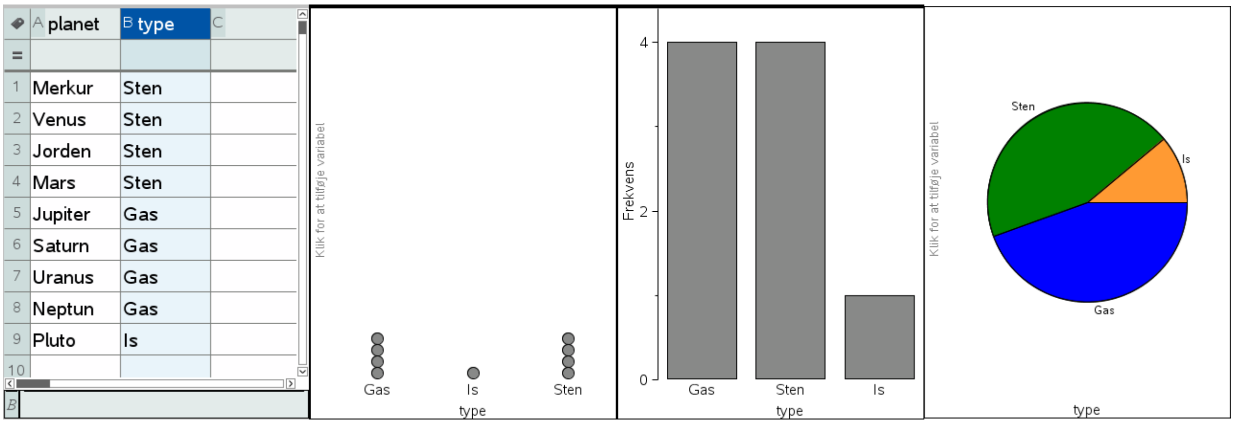
Standarddiagrammer

Standarddiagrammer kan tegnes som *hurtiggrafer*, idet vi indtaster værdierne for de involverede variable i **Lister og Regneark**-værkstedet, markerer dem (klik øverst i søjlebogstavet), højreklikker i markeringen og vælger **Hurtiggraf** fra **Data**-menuen.

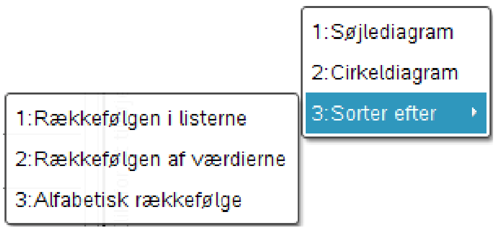
Standarddiagrammer kan også tegnes som almindelige diagrammer ved at tilføje et **Diagrammer og statistik**-værksted, og så klikke på den vandrette akse for at tilføje en variabel. Den pågældende variabel afbildes da netop som et standarddiagram.

1. Enkeltvariable

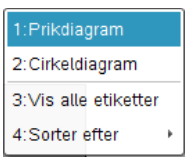
Hvis der er én **kategorisk variabel** afbildes den som standard i et *prikdiagram*, der efterfølgende kan konverteres til et *søjlediagram* eller et *cirkeldiagram* (ved at højre-klikke i grafvinduet). Her er det vist med planettyper, der kan være sten-planeter, gas-planeter eller is-planeter.

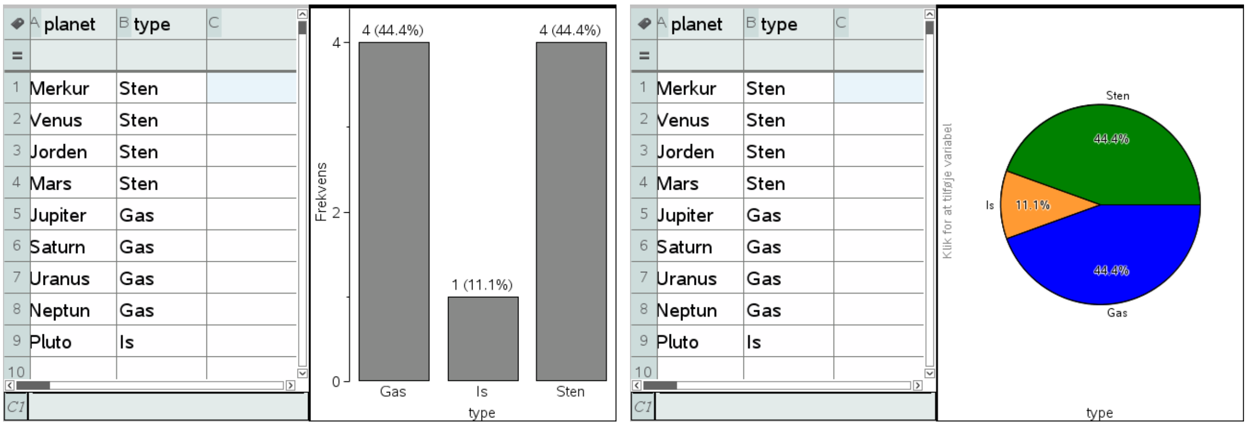


Der er en vis frihed i disse diagrammer. I prikdiagrammet og søjlediagrammet kan man frit trække i etiketterne på den vandrette akse og dermed ændre den rækkefølge de vises i. Man kan også højreklikke og *sortere etiketterne* efter rækkefølgen listerne (i dette tilfælde sten, gas og is), rækkefølgen af værdierne (i dette tilfælde gas med 4, sten med 4 og is med 1: Da der er to planettyper, der er lige hyppige sorteres de alfa-numerisk) og endelig alfanumerisk rækkefølge (gas, is og sten):



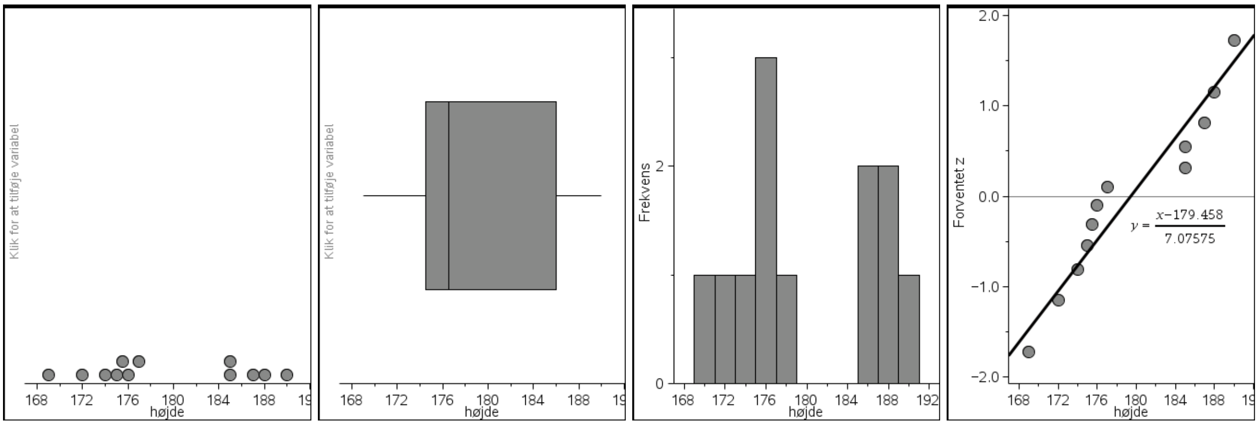
Endelig kan man få **Vist alle etiketter** for søjlediagrammet og cirkeldiagrammet:





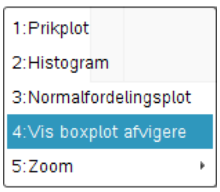
Hvis der er én **numerisk variabel** afbildes den som standard i et *prikdiagram*, der efterfølgende kan konverteres til et *histogram*, et *boksplot* eller et *normalfordelingsplot*. Her er det vist med højderne for drengene i en klasse.

Ali	172
Andreas	187
Christian	174
Jakob	175
Jamal	169
Julian	176
Kasper L	185
Kasper Q	190
Martin	175.5
Mohamed	185
Nikita	177
Paw	188



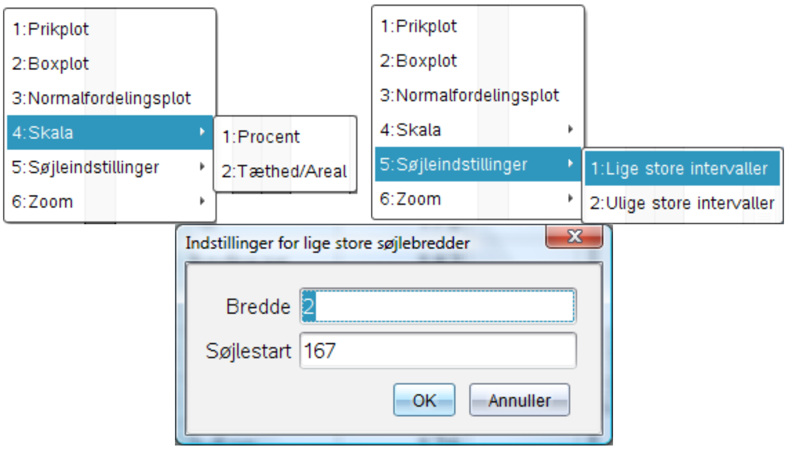
Normalfordelingsdiagrammet er det mest komplicerede af disse diagrammer og det vil blive diskuteret i et særskilt afsnit senere.

Der er en vis frihed i disse diagrammer. I boksplottet kan man vælge om man vil se afvigere/perifere observationer eller om man vil udstrække boksplotgrænserne til minimum og maksimum.



I vores tilfælde er der dog ingen afvigere ☺

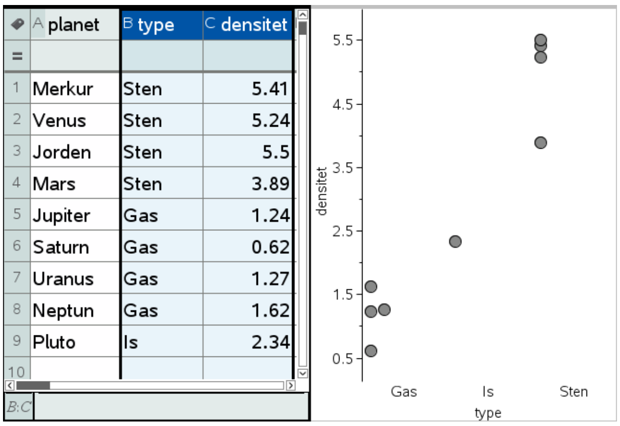
I histogrammet kan man frit vælge om man vil inddele førsteaksen efter lige store eller ulige store intervaller. Det sidste kræver dog, at man tilføjer en liste over intervalgrænser. Tilsvarende kan man vælge mellem forskellige optællingsskalaer: Frekvens/Hyppighed, Procent eller Tæthed/Areal. Det sidste svarer mest til det traditionelle histogram i undervisningen.



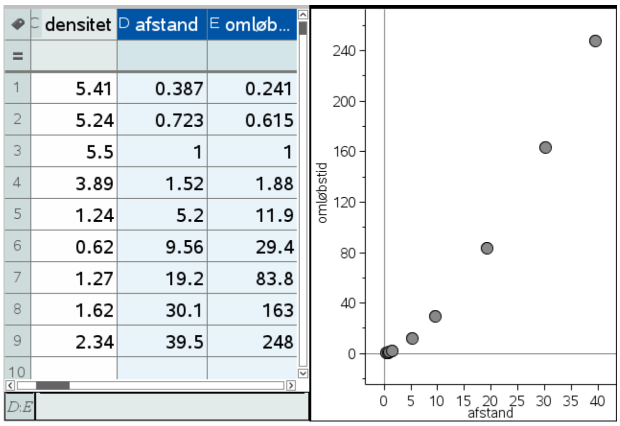
Men man kan selvfølgelig også justere histogrammets udseende ved simpelthen at trække i båsene og derved gøre dem tykkere/tyndere ☺

2. dobbeltvariable

Hvis vi har to variable i spil og vil fremstille et standarddiagram, der viser sammenhængen mellem de to variable afhænger diagramtypen naturligvis af hvilken type, de to variable tilhører. De to vigtigste typer er **kategorisk-numerisk** (dvs. den uafhængige variabel er kategorisk, mens den afhængige variabel er numerisk) og **numerisk-numerisk** (dvs. den uafhængige variabel er numerisk, ligesom den afhængige variabel er numerisk)

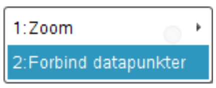


	A planet	B type	C densitet
1	Merkur	Sten	5.41
2	Venus	Sten	5.24
3	Jorden	Sten	5.5
4	Mars	Sten	3.89
5	Jupiter	Gas	1.24
6	Saturn	Gas	0.62
7	Uranus	Gas	1.27
8	Neptun	Gas	1.62
9	Pluto	Is	2.34



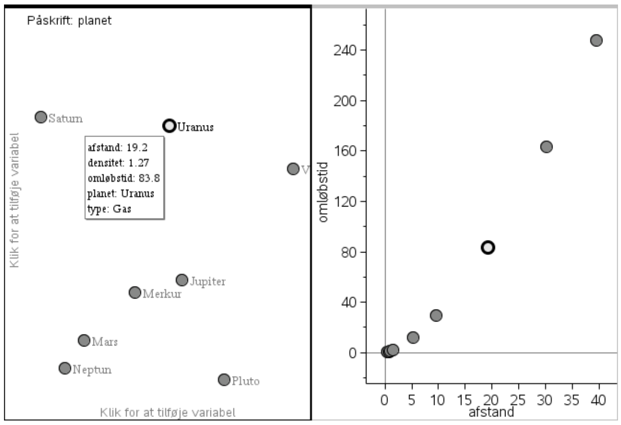
	C densitet	D afstand	E omløb...
1	5.41	0.387	0.241
2	5.24	0.723	0.615
3	5.5	1	1
4	3.89	1.52	1.88
5	1.24	5.2	11.9
6	0.62	9.56	29.4
7	1.27	19.2	83.8
8	1.62	30.1	163
9	2.34	39.5	248

I det første tilfælde afsættes prikdiagrammer for hver af etiketterne. Men de kan selvfølgelig omdannes til boksplots og histogrammer. I det andet tilfælde oprettes et punktplot. Hvis det ønskes kan man forbinde datapunkterne med linjestykker, men det er sjældent relevant



Desværre er det ikke muligt at få hæftet etiketter automatisk på datapunkterne i punktplottet. Som nødløsning kan man holde et ustruktureret diagram åbent ved siden af punktplottet, for så vil et klik på et datapunkt i et af diagrammerne automatisk koble det til det tilsvarende datapunkt i det andet diagram ☺

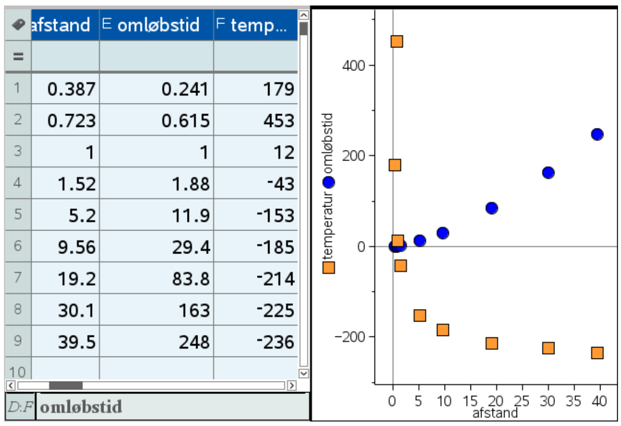
På den måde kan man hurtigt identificere et datapunkt og samtidigt få oplyst alle de tilhørende værdier i datasættet.



3. Multivariable

Hvis vi har mange variable i spil og vil fremstille et standarddiagram, der viser sammenhængen mellem disse variable afhænger diagramtypen naturligvis igen af hvilken type, de forskellige variable tilhører. Denne gang skal *alle* variablene være **numeriske**. Den første variabel håndteres da som den uafhængige variabel, mens de øvrige variable håndteres som afhængige variable:

Obs
Hvis man opretter et **Diagrammer og Statistik**-vindue og vil *tilføje flere variable* til en akse er det nødvendigt at højreklikke i aksefeltet for at få lov til at tilføje endnu en variabel. Ellers udskiftes variablen blot ☺



	afstand	omløbstid	temp...
1	0.387	0.241	179
2	0.723	0.615	453
3	1	1	12
4	1.52	1.88	-43
5	5.2	11.9	-153
6	9.56	29.4	-185
7	19.2	83.8	-214
8	30.1	163	-225
9	39.5	248	-236
10			

Normalfordelingsdiagrammer

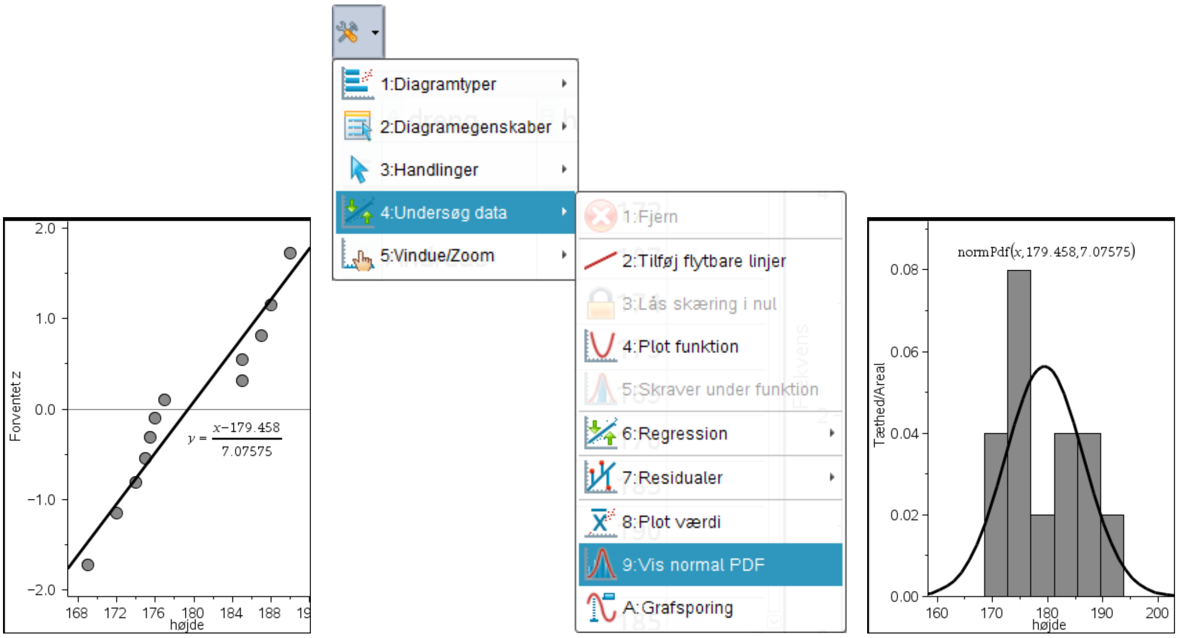
Der er to typer normalfordelingsdiagrammer, man kan knyttet til et numerisk datasæt: Dels normalfordelingsplottet, som vi allerede har set på, dels kan man overlejre et histogram med den tilsvarende normalfordelingsgraf ved at vælge menupunktet

4:Undersøg data ▸ 9:Vis normal PDF

I begge tilfælde udregnes *middelværdien* for datasættet og den såkaldte *stikprøvespredning* og der overlejres med den tilsvarende normalfordeling med samme middelværdi og spredning.

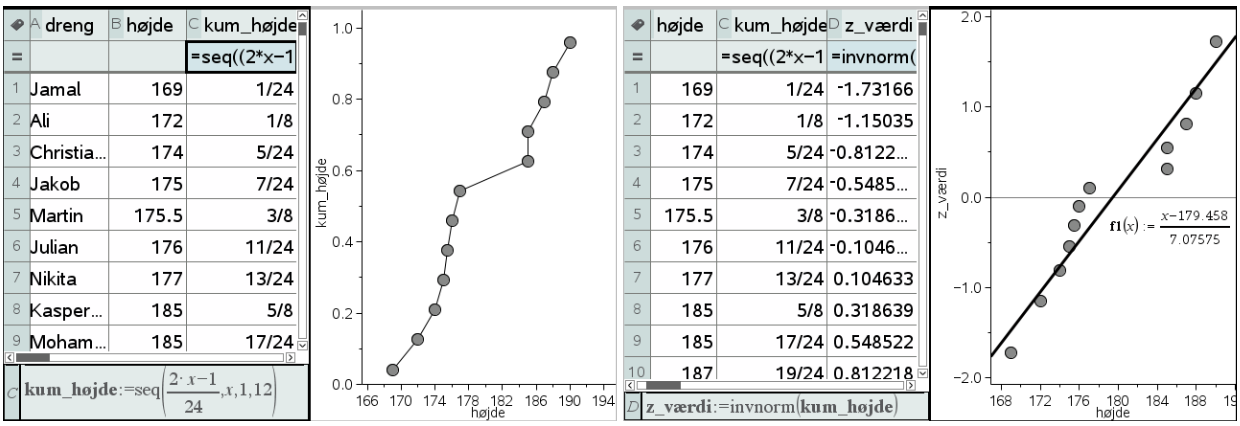
I normalfordelingsplottet er det den kumulerede fordeling, der afsættes på et standard normalfordelingspapir, hvor den kumulerede normalfordeling fremstår som en ret linje.

I histogramtilfældet er det tæthedsfordelingen/punktfordelingen, der afsættes sammen med histogrammet i den samme skala som histogrammet. Hvis man ønsker et klassisk diagram skal man derfor skifte til skalaen Tæthed/Areal:



Mens den klokkeformede kurve overlejret et histogram er rimelig nem at forstå, kan det være sværere at tolke normalfordelingsplottet. Det kan da være nyttigt, at vide lidt mere om hvordan det er fremkommet, selvom det følgende nødvendigvis bliver lidt teknisk ☺

Vi har et datasæt bestående af 12 højder. Vi sorterer først datasættet efter højde. Derefter tildeler vi ethvert datapunkt, dvs. enhver højde, et interval, der netop svarer til en tolvtedel af enhedsintervallet [0;1]; dvs. det første datapunkt tildeles intervallet [0;1/12], det andet tildeles intervallet [1/12;2/12] osv. Selve datapunktet knyttes til midtpunktet af intervallet, dvs. det første datapunkt knyttes til 1/24, det andet datapunkt til 3/24 osv. På basis af disse kumulerede frekvenser kan vi nu tegne sumkurven for datasættet:



Hvis der er tale om en normalfordeling vil sumkurven nu ligne den typiske S-formede kurve. Men da det kan være svært at sammenligne S-formede kurver, retter vi den i stedet op til en ret linje ved at benytte normalfordelingsaksen på andenaksen, dvs. ved at overføre den kumulerede fordeling til et standardnormalfordelingspapir. Det sker ved at erstatte de kumulerede frekvenser med de tilsvarende z-værdier for en standardnormalfordelingen, dvs. som vist transformere dem med den omvendte normalfordeling. Til sidst tilføjes den rette linje med forskriften

$$y = \frac{x - middelværdi}{stikprøvespredning}$$

Og vi har nu set hvordan standardnormalfordelingsplottet er fremkommet ☺

	A elev	B køn	C højde
=			
1	Ali	Dreng	172
2	Andreas	Dreng	187
3	Camilla	Pige	162
4	Cecilie	Pige	173
5	Christian	Dreng	174
6	Christina	Pige	163
7	Cilie	Pige	169.5
8	Heidi	Pige	169
9	Jakob	Dreng	175
10	Jamal	Dreng	169
11	Jane	Pige	166
12	Julian	Dreng	176
13	Kasper L	Dreng	185
14	Kasper O	Dreng	190
15	Louise K	Pige	165
16	Louise R	Pige	168.5
17	Maria	Pige	170.5
18	Martin	Dreng	175.5
19	Marzia	Pige	151
20	Mohamed	Dreng	185
21	Nikita	Dreng	177
22	Paw	Dreng	188
23	Rose	Pige	174.5
24	Sofie K	Pige	165
25	Sofie T	Pige	172.5

Diagrammer med kategoriske opdelinger: Sammenligning af fordelinger

Alle de diagrammer vi har kigget på hidtil er standarddiagrammer, som vi har kunnet tegne med hurtiggraf (eller bare oprettet et **Diagrammer og statistik**-værksted og tilføjet variable til akserne). Vi vil nu se lidt nærmere på nogle andre diagramtyper, der kræver lidt mere omtanke.

Vi lægger ud med *kategoriske splitdiagrammer*, hvor vi i første omgang splitter **numeriske data** efter **kategoriske data**. Her afhænger proceduren af, hvordan datasættet er fremstillet. Der findes nemlig mindst to forskellige måder at håndtere kategoriske variable på i forbindelse med numeriske data.

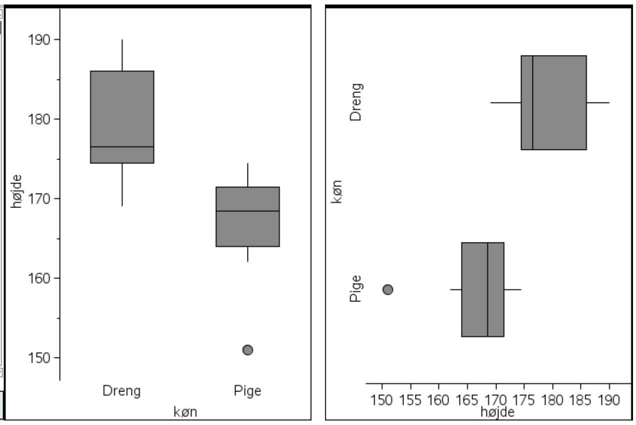
Vi vender tilbage til eksemplet med højdemålinger i en klasse, men inkluderer denne gang også pigerne. I et *velstruktureret* datasæt vil der nu være tre variable: En kategorisk variabel for eleverne, en kategorisk variabel for deres køn og en numerisk variabel for deres højde. Alle listerne er lige lange (med 25 elementer svarende til de 25 elever i klassen).

Afbilder vi nu højden som funktion af kønnet får vi netop den ønskede opdeling af højderne på de to køn og kan umiddelbart sammenligne de to fordelinger. Vi kan som vist også afbilde køn som funktion af højden, hvis det giver bedre overblik:

Obs
Standarddiagrammer kan frembringes typisk ved blot at vælge variable til aksefelterne. Men fx kategoriske opdelinger og kombinationsdiagrammer kræver typisk, at man højreklikker i aksefelterne for at få adgang til flere muligheder.

	elev	B køn	C højde
=			
1		Dreng	172
2	dreas	Dreng	187
3	ımilla	Pige	162
4	cilie	Pige	173
5	ıristian	Dreng	174
6	ıristina	Pige	163
7	ie	Pige	169.5
8	idi	Pige	169
9	kob	Dreng	175
10	mal	Dreng	169

B:C



Men det er desværre ikke alle datasæt, der er velstrukturerede. Typisk vil de to køn være skilt ud på separate lister (de er altså **ustakkede**)

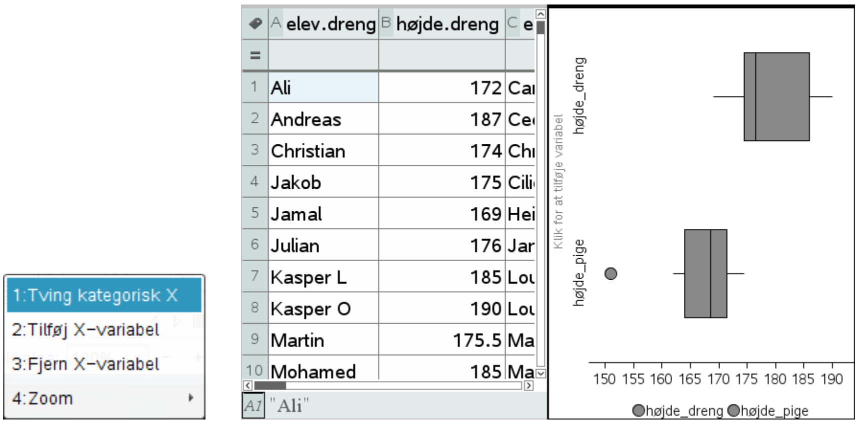
	A elev.dreng	B højde.dreng	C elev.pige	D højde.pige
=				
1	Ali	172	Camilla	162
2	Andreas	187	Cecilie	173
3	Christian	174	Christina	163
4	Jakob	175	Cilie	169.5
5	Jamal	169	Heidi	169
6	Julian	176	Jane	166
7	Kasper L	185	Louise K	165
8	Kasper O	190	Louise R	168.5
9	Martin	175.5	Maria	170.5
10	Mohamed	185	Marzia	151
11	Nikita	177	Rose	174.5
12	Paw	188	Sofie K	165
13			Sofie T	172.5

5: Deskriptiv statistik

Tip
Hvis du ønsker det kan du samle de to lister til en enkelt stakket liste over hele klassens højder ved at sammenkæde de to lister med **augment**-kommandoen.

Man har altså delt datasættet op i to separate datasæt, et for drenge og et for piger. Når man læser værdier på tværs hænger de derfor ikke længere sammen. Strengt taget burde de to datasæt derfor skilles ud på hvert sit regneark, men det er der trods alt ikke tradition for ☺

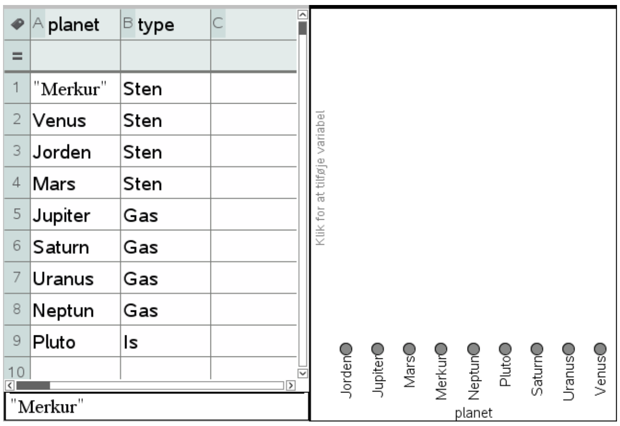
I stedet opretter man et **Diagrammer og Statistik**-vindue, hvor man først afsætter variablen **højde.dreng** og derefter højreklikker man i aksefeltet for at få lov til at tilføje en *X*-variabel (hvis man venstreklikker får man bare lov til at udskifte en *X*-variabel). Derefter tilføjes variablen **højde.pige**.



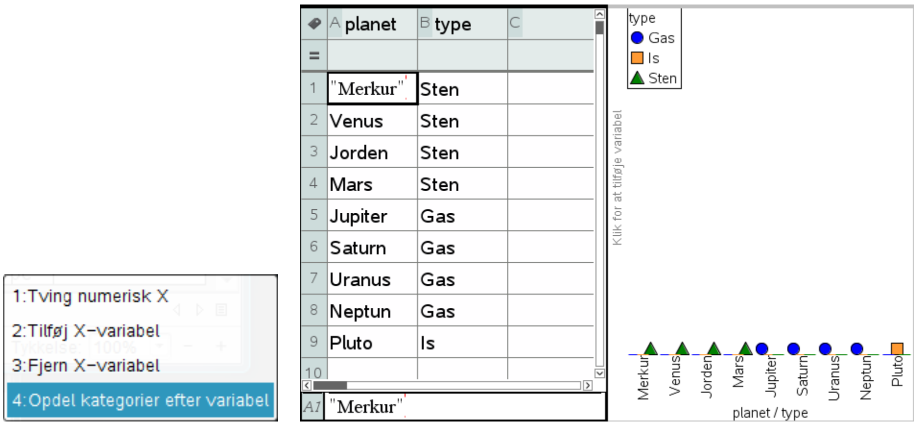
Denne gang kan man dog ikke bytte om på akserne, dvs. man kan *ikke* afsætte boksplottene lodret. Det er dog en lille pris at betale for at få lov til at holde drenge og piger adskilt ☺

Vi vender os derefter mod tilfældet, hvor vi ønsker at afbilde sammenhængen mellem **to kategoriske variable**, dvs. vi opdeler nu én kategorisk variabel efter en anden **kategorisk variabel**.

Vi lægger ud med et meget simpelt eksempel, nemlig planeternes opdeling på type ("Sten", "Gas" og "Is"). Vi opretter da først et **Diagrammer og Statistik**-vindue, hvor vi afsætter variablen planet:

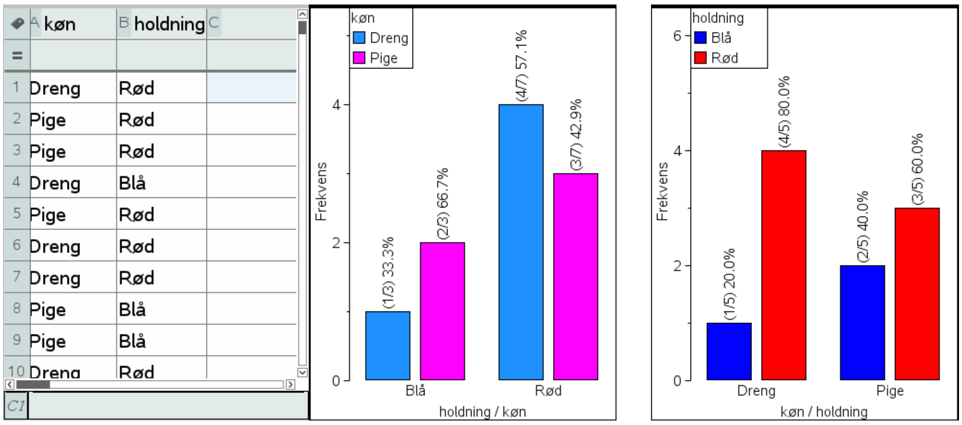


Derefter højreklikker vi i aksefeltet og vælger **Opdel kategorier efter variabel**. Det giver mulighed for at tilføje variablen **type**, så der til hver planet nu knyttes en *signatur*, der tydeligt viser hvilken type, der er tale om, idet regnearket vælger signaturen ● for Gas, signaturen ■ for Is og endelig signaturen ▲ for Sten.



Men langt den vigtigste anvendelse af opdeling efter kategorier stammer fra optælling af krydstabeller. Vi tænker os at vi har to kategoriske variable, der typisk stammer fra et spørgeskema som fx registrerer den adspurgtes **køn** og **politisk holdning**. Fx kan vi se på et meget simpelt eksempel med 10 spørgeskemaer, hvor det registreres om man tilhører *rød blok* eller *blå blok* i folketinget.

Vi kan da enten afsætte **holdning** først og derefter opdele efter **køn**, eller vi kan afsætte køn først og derefter opdele efter holdning. I begge tilfælde har vi skiftet til et **søjlediagram**, slået **Vis alle etiketter** til og farvelagt søjlerne passende:



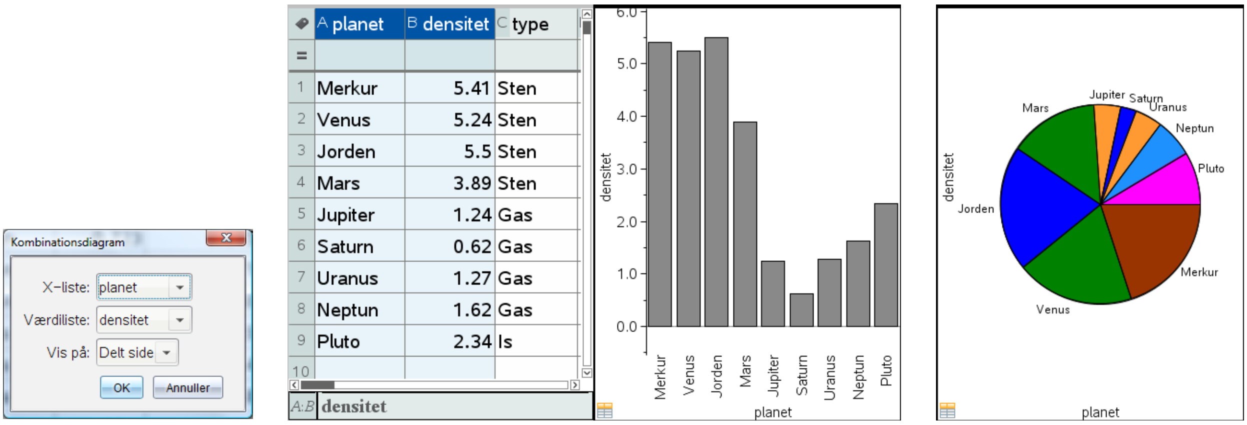
Vi får da ikke bare optalt hvor mange der findes i hver af de krydsede kategorier, fx at der findes netop 2 piger, der stemmer på blå blok. Vi får også de procentuelle fordelinger på henholdsvis holdning og køn. Fx viser det første diagram at 1/3 af stemmerne på blå blok kommer fra drengene, mens de resterende 2/3 af stemmerne kommer fra pigerne. Tilsvarende viser det andet diagram at 20% af drengene stemmer på blå blok, mens de resterende 80% af drengene stemmer på rød blok osv.

Men måske langt den vigtigste anvendelse af et sådant såkaldt **grupperet søjlediagram** er nok at man uden videre kan udfylde en **krydstabel/pivottabel** ved aflæsning af hyppighederne i diagrammerne. De kan så udfyldes på to ledder alt efter hvilken variabel man afsætter lodret og hvilken vandret.

Holdning\køn	Dreng	Pige	I alt
Blå	1	2	3
Rød	4	3	7
I alt	5	5	10

Kombinationsdiagrammer

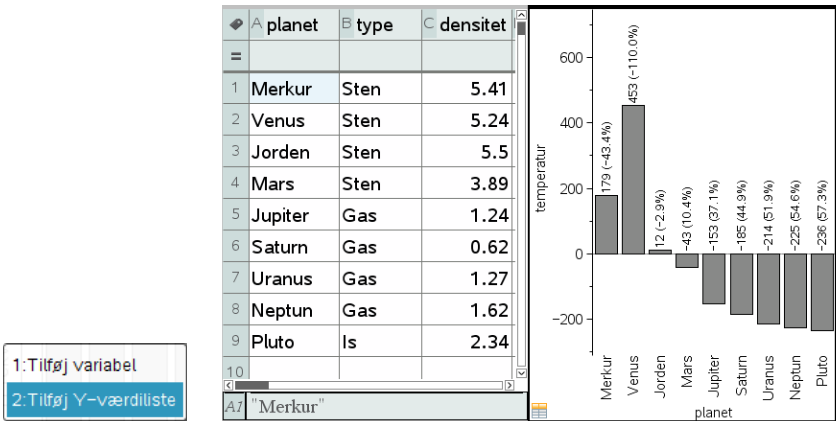
Vi kommer så endelig til den sidste type diagram, hvor man kombinerer en variabel med en **værdiliste**, og får afsat værdierne som søjler (med vilkårligt fortegn) eller som cirkeldiagram (hvis alle værdierne er positive). Da vi kombinerer en kategorisk/numerisk variabel med en numerisk variabel kaldes det et **kombinationsdiagram**. Her er det fx først vist med planeters densiteter, hvor vi har markeret søjlerne med planeternes navne og densiteter og derefter højreklikket og valgt kombinationsdiagram i **Data**-menuen. Da densiteterne er positive kan vi både fremstille et *søjlediagram* og et *cirkeldiagram*



A planet	B densitet	C type
Merkur	5.41	Sten
Venus	5.24	Sten
Jorden	5.5	Sten
Mars	3.89	Sten
Jupiter	1.24	Gas
Saturn	0.62	Gas
Uranus	1.27	Gas
Neptun	1.62	Gas
Pluto	2.34	Is

Læg mærke til ikonet nederst til venstre i diagrammet. Det markerer netop, at der er tale om et kombinationsdiagram.

Derefter har vi opbygget et *kombinationsdiagram* over planeternes temperaturer ved at oprette et **Diagrammer og statistik**-vindue, tilføjet variablen **planet** på førsteaksen og derefter højreklikket i aksefeltet for anden-aksen, hvilket giver mulighed for at vælge **Tilføj Y-værdiliste**, som netop anvendes til at frembringe et kombinationsdiagram for temperaturen som funktion af planeten:



A planet	B type	C densitet
Merkur	Sten	5.41
Venus	Sten	5.24
Jorden	Sten	5.5
Mars	Sten	3.89
Jupiter	Gas	1.24
Saturn	Gas	0.62
Uranus	Gas	1.27
Neptun	Gas	1.62
Pluto	Is	2.34

Denne gang er nogle af temperaturerne negative og man kan derfor *ikke* vælge et cirkeldiagram ☺

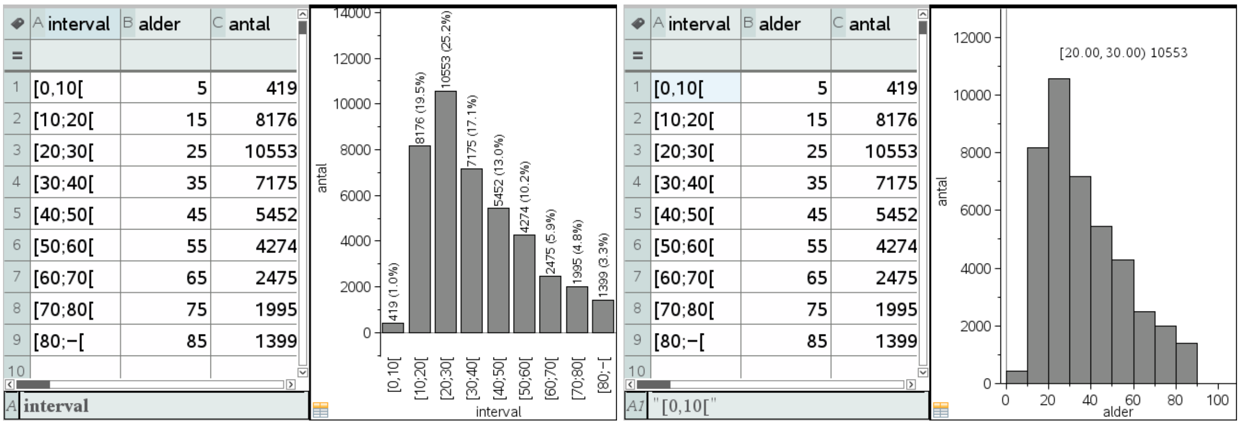
En anden meget vigtig anvendelse af kombinationsdiagrammer er i forbindelse med de såkaldte **hyppighedstabeller**, der fx optræder i forbindelse med grupperede observationer, hvor vi ikke har umiddelbar adgang til de rå data, men kun til en optælling af hyppighederne for de enkelte kategorier eller intervaller.

I så fald er det hyppighedslisten, der fungerer som værdiliste i kombinationsdiagrammet.

Hvis der fx er oplyst en tabel over forekomsten af kriminalitet fordelt på ofrets alder opdelt efter aldersgrupper i befolkningen:

Ofrets alder (interval)	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-
Antal (hyppighed)	419	8176	10553	7175	5452	4274	2475	1995	1399

Vi kan da indskrive tabellen i et regneark på to forskellige måder: Vi kan holde intervallerne som en kategorisk variabel eller vi kan omdanne intervallerne til en numerisk variabel. Det nemmeste er da at antage at alle intervallerne er lige lange, selvom vi strengt taget ikke ved noget om det sidste interval, og så anvende intervallets midtpunkt, som den karakteristiske værdi:



I det første tilfælde afbildes tabellen som et søjlediagram, fordi den uafhængige variabel er en kategorisk variabel. I det andet tilfælde afbildes tabellen som et histogram, fordi den uafhængige variabel er en numerisk variabel. I det sidste tilfælde har vi efterfølgende justeret båsene, så de har bredden 10 og starter i 0 ☺

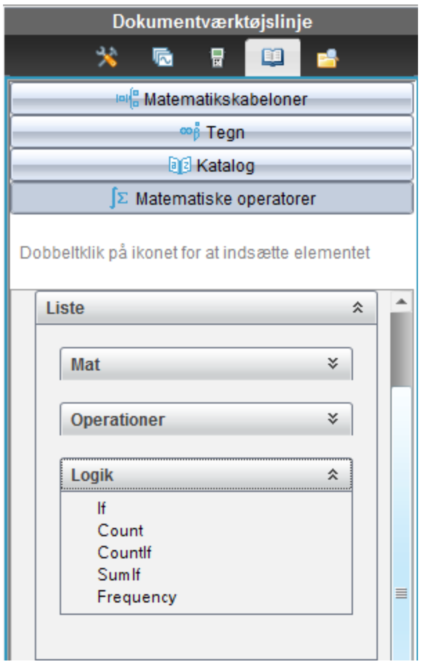
Beregninger hørende til deskriptiv statistik

Vi har nu set hvordan vi kan tegne fordelingerne for kategoriske og numeriske variable i passende diagrammer: prikdiagrammer, søjlediagrammer og cirkeldiagrammer for de kategoriske variable samt prikdiagrammer, boksplots og histogrammer for de numeriske variable. De kan så udvides til grupperede søjlediagrammer for to kategoriske variabel og punktplots for to numeriske variable.

Vi kigger nu nærmere på hvordan vi kan regne på fordelingerne:

Optællinger af kategoriske variable

Ved håndtering af kategoriske (og numeriske data) får vi brug for diverse *betingede kommandoer*, til at tælle hyppigheden for kategorier, grupperede datasæt osv. De vigtigste kommandoer er samlet i **Logik**-kommandoerne for lister (dvs. de betingede kommandoer) under **Matematiske operatorer** i venstre sidepanel



Det drejer sig om den betingede IF-kommando, der skrives **ifFn**, for ikke at forveksle den med programmerings-kommandoen **IF**, om tællekommandoerne **Count** og **CountIf**, om sum-kommandoerne **Sum** og **SumIf**, og endelig om **Frequency**-kommandoen, der optæller hyppigheder.

Her er nogle typiske anvendelser, hvor vi igen ser på datasættet fra planetsystemet. Vi kan fx bruge **CountIf**-kommandoen til at tælle, hvor mange planeter, der er af hver type, dvs. vi skal have frembragt en hyppighedstabel. Vi indskriver da cellekommandoen

$$\boxed{D1} = \text{CountIf}(\textbf{type}, \text{C1})$$

Efterfølgende trækkes kommandoen ned gennem tabellen (gult område):

	A planet	B type	C kategori	D antal	E densitet	F afstand	G om
=							
1	Merkur	Sten	Sten	4	5.41	0.387	
2	Venus	Sten	Gas	4	5.24	0.723	
3	Jorden	Sten	Is	1	5.5	1	
4	Mars	Sten			3.89	1.52	
5	Jupiter	Gas			1.24	5.2	
6	Saturn	Gas			0.62	9.56	
7	Uranus	Gas			1.27	19.2	
8	Neptun	Gas			1.62	30.1	
9	Pluto	Is			2.34	39.5	
10							

D1 =countif(**type**,*c1*)

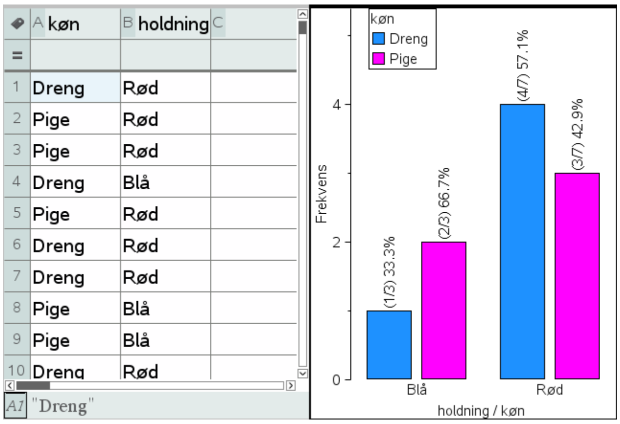
	A planet	B type	C kategori	D antal	E densitet	F afstand	G om
=				=frequenc			
1	Merkur	Sten	Gas	4	5.41	0.387	
2	Venus	Sten	Is	1	5.24	0.723	
3	Jorden	Sten	Sten	4	5.5	1	
4	Mars	Sten		0	3.89	1.52	
5	Jupiter	Gas			1.24	5.2	
6	Saturn	Gas			0.62	9.56	
7	Uranus	Gas			1.27	19.2	
8	Neptun	Gas			1.62	30.1	
9	Pluto	Is			2.34	39.5	
10							

D **antal**:=frequency(**type**,**kategori**)

Bruger vi i stedet **Frequency**-kommandoen, skal kategorierne først ordnes alfanumerisk, da **Frequency**-kommandoen tæller, hvor mange data, der går forud for kategorien. Ydermere får vi en ekstra optælling af de elementer, der ikke er dækket ind af kategorierne. Det er en arv fra Excel, der fungerer på samme måde. Man skal derfor typisk fjerne det overskydende 0, hvis man vil tegne og regne på hyppighedstabellen. På grund af denne skavank er derfor ofte at foretrække at bruge **CountIf**-kommandoen i stedet for **Frequency**-kommandoen ☺

Krydstabeller/Pivottabeller*

TI-Nspire CAS understøtter desværre ikke pivottabeller direkte. Som vi har set kan man nemt oprette et grupperet søjlediagram, hvoraf tallene i pivottabellen fremgår direkte. Men hvis vi skal udregne dem selv kræves der lidt behændighed ☺ Vi tager igen udgangspunkt i eksempel med køn og holdninger for 10 elever:



Af hensyn til senere tegninger/udregninger kan det betale sig at flytte krydstabellen op over stregen, dvs. bruge de vandrette kategorier **rød** og **blå** som variabelnavne. Vi bruger da cellekommandoen

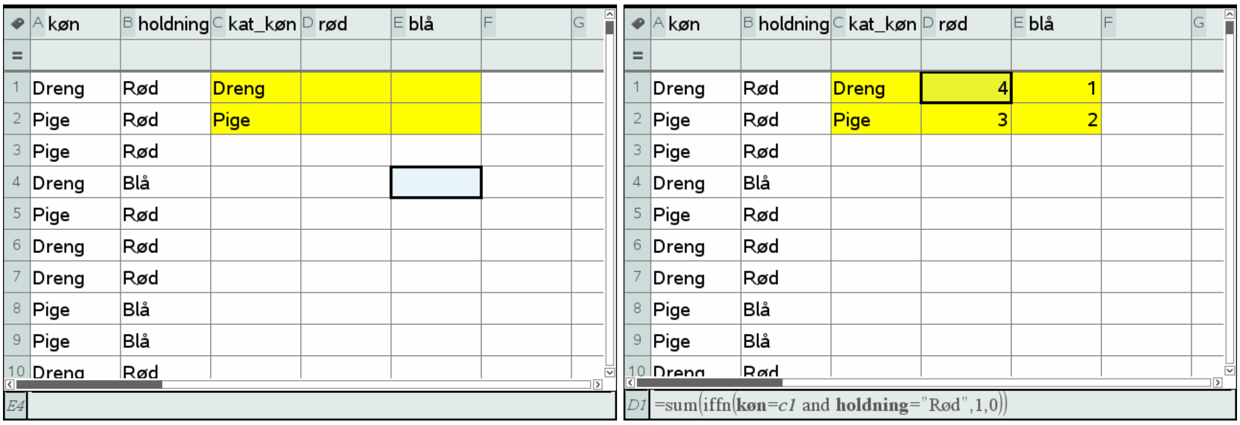
$$\boxed{D1}\ =\ \text{sum}(\text{ifFn}(\textbf{køn} = C1 \text{ and } \textbf{holdning} = \text{"Rød"},1,0))$$

til at finde antallet af *røde drenge*. Den trækkes derefter ned gennem listen for **rød**. Her bevirker **ifFn**-kommandoen at der fremstilles en liste med 1-taller og 0-taller, idet den får værdien 1, hver gang der er en rød dreng ud for hinanden i listerne **køn** og **holdning**. Summen af denne liste angiver netop antallet af røde drenge.

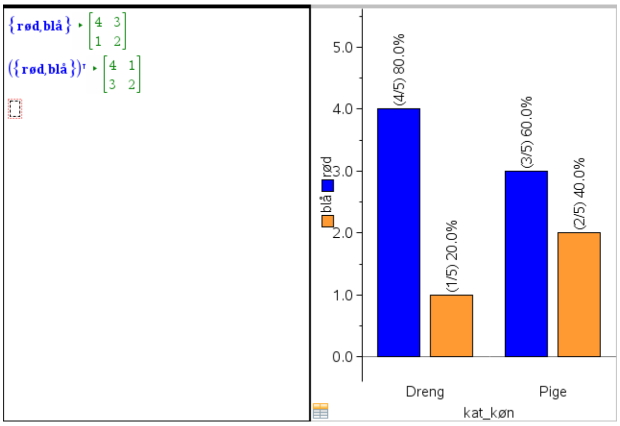
Derefter bruges cellekommandoen

$$\boxed{D1}\ =\ \text{sum}(\text{ifFn}(\textbf{køn} = C1 \text{ and } \textbf{holdning} = \text{"Blå"},1,0))$$

til at finde antallet af *blå drenge*. Den trækkes tilsvarende ned gennem tabellen og pivottabelen er færdig.

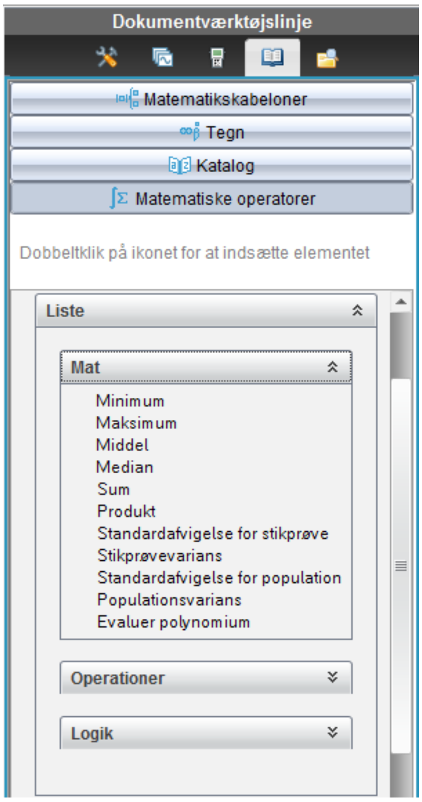


Netop fordi vi har flyttet krydstabellen op over stregen er det nu nemt at referere til den som en krydstabel/matrix ved bare at sætte krøllede parenteser yderst, dvs. skrive {**rød**,**blå**} (eventuelt med en transponeringskommando for at listerne skrives lodret ligesom i regnearket). Det er bekvemt, hvis vi fx skal udføre en uafhængighedstest, men mere om dette i kapitel 7. Tilsvarende er det nemt at frembringe et grupperet søjlediagram ved først at afsætte variablen **kat_køn** ud af første aksen og derefter tilføje variablen **rød** og **blå** som *værdi-lister*, dvs. vi fremstiller et kombinationsdiagram.



Enkeltvariabelstatistik for numeriske variable

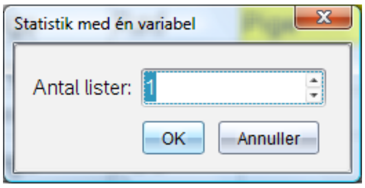
Ved håndtering af lister med numeriske data får vi tilsvarende brug for matematik-kommandoerne under **Liste** i **Matematiske operatorer**

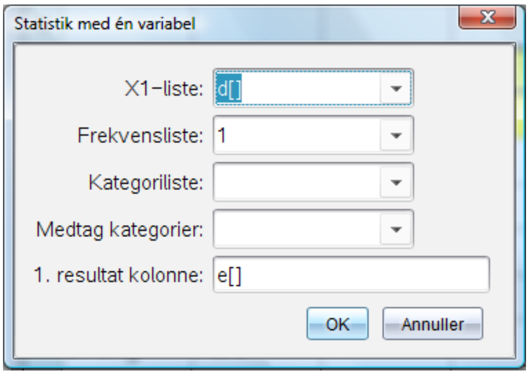


Her finder vi de fleste statistiske deskriptorer (dog *ikke* fraktil- og kvartil-kommandoer, hvilket bl.a. skyldes en uheldig definition af kvartiler i TI-Nspire CAS, der ikke kan udvides til vilkårlige fraktiler). Det er vigtigt, at bemærke at de vigtigste deskriptorer accepterer *to argumenter*, så de *også* kan bruges til at regne på værdilister i form af fx hyppighedslister eller sandsynlighedsfordelinger.

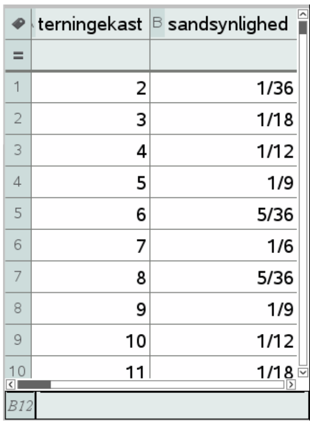
Middelværdi:	`mean(Liste [,frekvListe])`
Median:	`median(Liste [,frekvListe])`
Stikprøvespredning:	`stDevSamp(Liste [,frekvListe])`
Stikprøvevarians:	`varSamp(Liste [,frekvListe])`
Populationsspredning:	`stDevPop(Liste [,frekvListe])`
Populationsvarians:	`varPop(Liste [,frekvListe])`

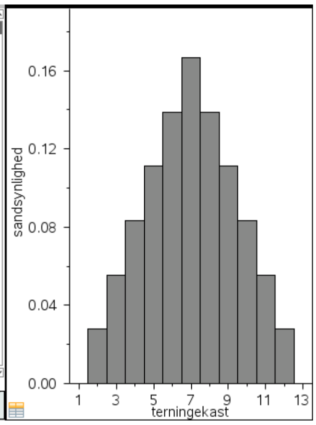
Men typisk vil man bruge **Enkeltvariabelstatistik**-kommandoen fra **Statistik**-menuen til at udregne alle deskriptorerne på én gang. Der fremkommer da en dialog-boks, der spørger hvor mange data-lister man vil udregne **statistik med én variabel** for (hvis datalisten er tilknyttet en værdiliste, eksempelvis en frekvensliste, tæller det stadigvæk kun som én dataliste):

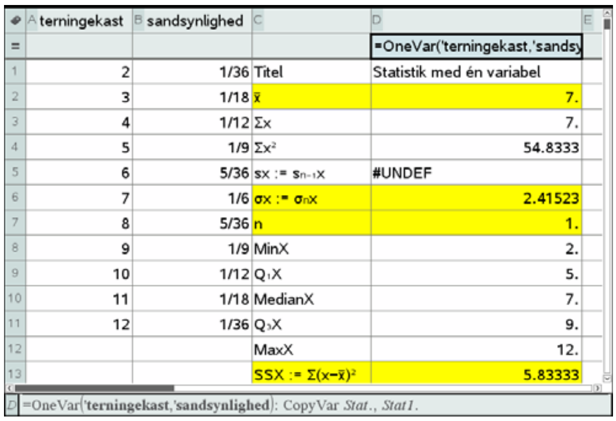




Her kan man nu tilføje en **data-liste** (X1-listen), en **værdi-liste** (Frekvenslisten, der som standard er sat til 1, dvs. alle data vægtes lige meget), samt en **Kategori-liste**, hvis man ønsker at opdele data efter kategorier, fx køn, dvs. udregne de såkaldte **sub-totaler**. Typisk ignorerer man denne, men hvis man har et datasæt, der tillader opdeling på kategorier, kan man oprette lister over de kategorier, som man ønsker medtaget. I første omgang ser vi dog kun på betydningen af frekvenslisten. Hvis vi fx har en sandsynlighedsfordeling, her summen af to terningekast, kan vi altså inkludere sandsynlighedslisten i beregningen, og der igennem få udregnet de vigtigste de-skriptorer sådom *forventningsværdien* $\bar{x}$,*spredningen* σx, *summen af sandsynlig-hederne* n og *variansen* $SSX := \Sigma(x-\bar{x})^2$ for sandsynlighedsfordelingen:



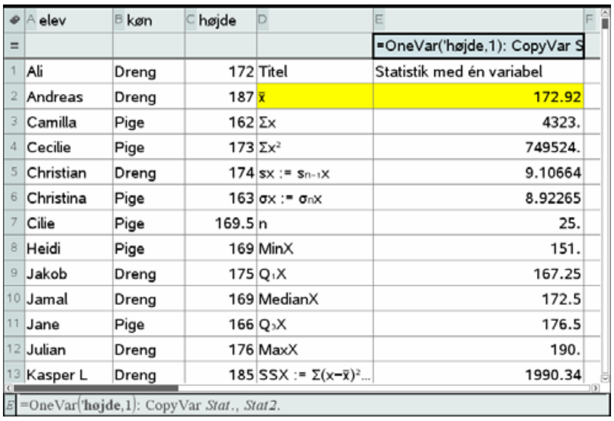




Udregning af subtotaler*

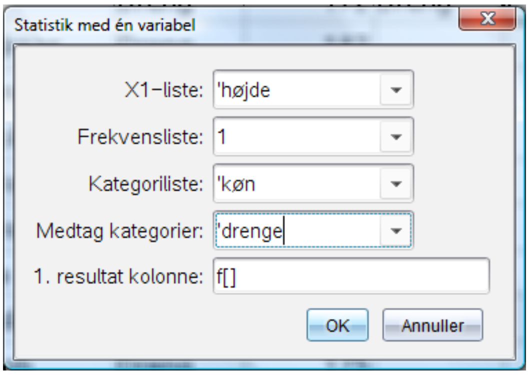
Vi vender os derefter mod opdeling efter kategorier. Vi bruger igen eksemplet med højderne i en klasse. Hvis vi nu udregner statistik for en variabel på højderne finder vi gennemsnitshøjden for hele klassen, dvs. 172.9 cm. Men så har vi jo ikke taget hensyn til, at der er stor forskel på drenges og pigers højder ☺

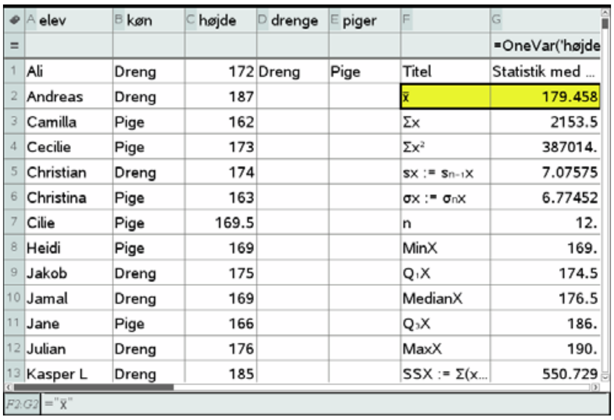
Hvis vi vil udregne subtotalerne, dvs. gennemsnitshøjden for drenge og piger hver for sig, må vi derfor indføre to nye lister, **drenge** og **piger**, som hver for sig rummer de kategorier, der hører under **drenge** henholdsvis **piger**. I vores tilfælde rummer listen **drenge** derfor kun en enkelt kategori, nemlig ”Dreng”, og tilsvarende rummer listen **piger** kun en enkelt kategori, nemlig ”Pige”.



	A elev	B køn	C højde	D	E
=					=OneVar('højde,1): CopyVar S
1	Ali	Dreng	172	Titel	Statistik med én variabel
2	Andreas	Dreng	187	$\bar{x}$	172.92
3	Camilla	Pige	162	Σx	4323.
4	Cecilie	Pige	173	Σx^2	749524.
5	Christian	Dreng	174	$sx := s_{n-1}x$	9.10664
6	Christina	Pige	163	$\sigma x := \sigma_n x$	8.92265
7	Cilie	Pige	169.5	n	25.
8	Heidi	Pige	169	MinX	151.
9	Jakob	Dreng	175	Q_1X	167.25
10	Jamal	Dreng	169	MedianX	172.5
11	Jane	Pige	166	Q_3X	176.5
12	Julian	Dreng	176	MaxX	190.
13	Kasper L	Dreng	185	$SSX := \Sigma(x-\bar{x})^2$...	1990.34

E =OneVar('højde,1): CopyVar *Stat.*, *Stat2.*





	A elev	B køn	C højde	D drenge	E piger	F	G
=							=OneVar('højde
1	Ali	Dreng	172	Dreng	Pige	Titel	Statistik med ...
2	Andreas	Dreng	187			$\bar{x}$	179.458
3	Camilla	Pige	162			Σx	2153.5
4	Cecilie	Pige	173			Σx^2	387014.
5	Christian	Dreng	174			$sx := s_{n-1}x$	7.07575
6	Christina	Pige	163			$\sigma x := \sigma_n x$	6.77452
7	Cilie	Pige	169.5			n	12.
8	Heidi	Pige	169			MinX	169.
9	Jakob	Dreng	175			Q_1X	174.5
10	Jamal	Dreng	169			MedianX	176.5
11	Jane	Pige	166			Q_3X	186.
12	Julian	Dreng	176			MaxX	190.
13	Kasper L	Dreng	185			$SSX := \Sigma(x...$	550.729

F2 G2 ="$\bar{x}$"

Gennemsnitshøjden for drenge er altså 179.5 cm. På samme måde finder man gennemsnitshøjden for piger.

Grupperede observationer

Ved meget store datasæt, mere end et par tusinde data, kan det være bekvemt at gruppere dataene og fremstille dem via en *hyppighedstabel*. Fx er det ikke praktisk at have adgang til de rå data for hele Danmarks befolkning. Det er langt mere praktisk at opdele Danmarks befolkning i eksempelvis aldersgrupper, og så angive hvor mange der befinder sig i hver gruppe.

Vi ser igen på en kriminalitetsstatistik

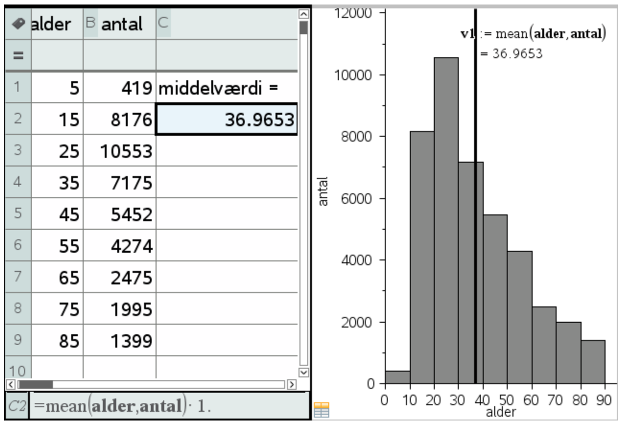
Ofrets alder (interval)	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-
Antal (hyppighed)	419	8176	10553	7175	5452	4274	2475	1995	1399

Hvordan kan vi nu udregne passende værdier for middelværdi, median og kvartiler? Der er alt for mange data til at vi kan arbejde med rådata, og desuden kender vi ikke den præcise fordeling af data indenfor de enkelte intervaller. Vi bygger derfor på en antagelse om at *data indenfor de enkelte aldersgrupper er fordelt jævnt*. De 419 ofre for kriminalitet i aldersgruppen 0-9 år har derfor en middelalder på 5 år osv. Ved udregning af middelværdien er det derfor bekvemt at opfatte ofrets alder som en numerisk variabel, der repræsenteres af *intervalmidtpunktet*. Vi opretter derfor den følgende tabel i et **Lister og Regneark**-værksted med intervalmidtpunkterne som første søjle og hyppighederne som anden søjle.

Vi kan da nemt udregne middelværdien som det vægtede gennemsnit af variablen **alder**, idet vi vægter med variablen **antal**, dvs. vi bruger cellekommandoen

= mean(**alder**, **antal**)

Som det ses kan middelværdien også nemt afsættes i det tilhørende histogram ved hjælp af kommandoen **Plot værdi** fra **Undersøg data**-menuen.



Men den samme teknik kan *ikke* brugs til at skønne over spredningerne, fordi vi jo netop *ikke* har fået fordelt dataene jævnt i de enkelte aldersgrupper – i stedet har vi samlet alle dataene i den enkelte gruppe i midtpunktet af intervallet. Af samme grund kan teknikken heller *ikke* bruges til at udregne et skøn over medianen.

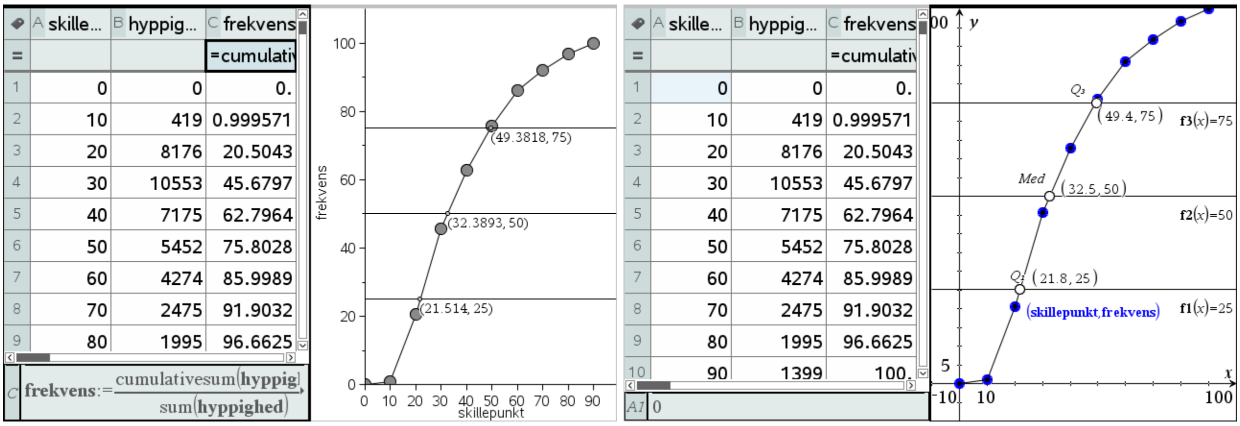
Vi er derfor nødt til også at kigge på *sumkurven* hørende til datasættet. Den fremstilles nemmest ud fra intervalgrænserne (som der er én mere af end antallet af intervaller), idet vi udvider listen over hyppigheder med et 0 til at begynde med, da der ingen data findes forud for det første intervalendepunkt. Vi opretter altså en søjle med skillepunkter og en søjle med hyppigheder. Søjlen med hyppigheder omdannes efterfølgende til en søjle over de relative frekvenser ved hjælp af kommandoen

$$\frac{\text{cumSum}(\textbf{hyppighed})}{\text{sum}(\textbf{hyppighed})} \cdot 100.$$

Derefter er vejen åben for at frembringe en sumkurve. Det kan gøres i et **Diagrammer og Statistik**-vindue, men så kan vi kun *aflæse* skæringspunkterne med de vandrette linjer $y = 25$, $y = 50$ og $y = 75$ (oprettet som grafer for konstante funktioner), fx ved at spore graferne (og trykke **ENTER**, når vi synes vi er tæt nok på skæringspunkterne).

Det er derfor bedre at oprette sumkurven i et **Graf**-vindue som et **punktplot**, forbinde punkterne med linjestykker og derefter bestemme skæringspunkterne med værktøjet **Skæringspunkter** fra **Punkt og Linje**-menuen under **Geometri**-menuen i **Graf**-værkstedet.

Vi finder da nemt såvel skøn over såvel kvartilerne som medianen. Da der netop er tale om skøn er der dog ingen grund til at sætte antallet af decimaler op ☺



Stykvis konstant og lineær interpolation i avanceret deskriptiv statistik*

Hvis vi vil *regne* i stedet for at *tegne* er vi nødt til at håndtere histogrammet og sumkurven som *grafer for fordelingsfunktioner*. Det er ikke helt simpelt, så det følgende afsnit forudsætter bl.a. fortrolighed med differential- og integralregning ☺

Obs
Husk at bruge kantede parenteser til at angive et element i en liste. Det første element i listen hedder derfor **liste**[1], det andet element **liste**[2] osv.

Vi starter med histogrammet, der er graf for en *stykvis konstant funktion*. Som udgangspunkt er definitionsmængden $[0;90[$ opdelt i 9 lige store intervaller med længde 10. Vi nummerer disse 9 intervaller fra 1 til 9. En given x-værdi hører derfor til intervallet med indekset

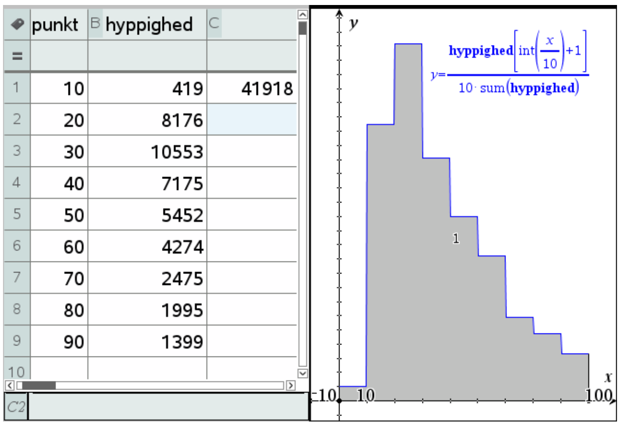
$$i = \mathrm{int}\left(\frac{x}{10}\right) + 1$$

Den tilhørende y-værdi er derfor givet ved

$$y = \frac{\mathbf{hyppighed}[\mathrm{int}(\frac{x}{10}) + 1]}{10 \cdot \mathrm{sum}(\mathbf{hyppighed})}$$

Her har vi divideret med $10 \cdot \mathrm{sum}(\mathbf{hyppighed})$, for at det samlede areal netop bliver 1 (idet faktoren 10 kommer fra Δx).

Tip
Når du skal udregne et integral ved hjælp af **Undersøg Graf**-menuen skal du først udpege den nedre grænse og dernæst den øvre grænse. Det kan gøres grafisk, men det er nemmere at indtaste værdierne direkte fra tastaturet.



Men så kan middelværdien jo netop udregnes som integralet

$$\bar{x} = \int_a^b x \cdot f(x)\,dx$$

hvor $f(x)$ er *tæthedsfunktionen*. Tilsvarende kan variansen udregnes som integralet

5: Deskriptiv statistik

$$\mathrm{Var}[x] = \int_a^b \left(x - \bar{x}\right)^2 \cdot f(x)\,dx$$

Vi finder da de følgende skøn over middelværdien, variansen og spredningen.

Tæthed:

$\mathbf{f}(x) := \dfrac{\mathbf{hyppighed}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]}{10 \cdot \mathrm{sum}(\mathbf{hyppighed})}$ ▸ *Udført*

Middelværdi:

$\mu := \int_0^{90} (x \cdot \mathbf{f}(x))\,dx$ ▸ 36.9653⚠

Varians:

$v := \int_0^{90} \left((x-\mu)^2 \cdot \mathbf{f}(x)\right)dx$ ▸ 383.177⚠

Spredning:

$\mathbf{sigma} := \sqrt{v}$ ▸ 19.5749

Vi har tidligere fundet middelværdien med elementære metoder, men nu får vi altså spredningen med i købet ☺

Vi vender os herefter mod *sumkurven*. Denne er stykvis lineær og fremkommer ved en *lineær interpolation* i tabellen over de kumulerede frekvenser. Vi finder *indekserne* for intervalendepunkterne på samme måde som før, idet vi denne gang tager udgangspunkt i tabellen med de 10 skillepunkter hørende til de 9 intervaller:

$$i_1 = \mathrm{int}\left(\frac{x}{10}\right) + 1, \quad i_2 = \mathrm{int}\left(\frac{x}{10}\right) + 2$$

Den tilhørende y-værdi er da givet ved formlen for lineær interpolation

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1)$$

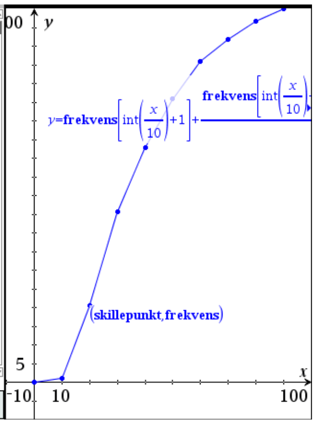
$$= \mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right] + \frac{\mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+2\right] - \mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]}{10} \cdot \left(x - \mathbf{skillepunkt}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]\right)$$

Vi finder derfor den følgende graf svarende til sumkurven. Vi har indlagt punktplottet sammen med grafen, så man kan se, at der netop er tale om sumkurven hørende til punktplottet ☺

Vi kan da som vist finde kvartiler og medianer ved at løse ligninger:

	A skille...	B hyppig...	C frekvens
=			=cumulativ
1	0	0	0.
2	10	419	0.999571
3	20	8176	20.5043
4	30	10553	45.6797
5	40	7175	62.7964
6	50	5452	75.8028
7	60	4274	85.9989
8	70	2475	91.9032
9	80	1995	96.6625
10	90	1399	100.

B1 0



Sumkurve:

$\mathbf{s}(x) := \mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right] + \dfrac{\mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+2\right] - \mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]}{10}\left(x - \mathbf{skillepunkt}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]\right)$

▸ *Udført*

Første kvartil:

$\mathrm{solve}(\mathbf{s}(x)=25, x=20)$ ▸ $x=21.7857$⚠

Median:

$\mathrm{solve}(\mathbf{s}(x)=50, x=30)$ ▸ $x=32.524$⚠

Tredje kvartil:

$\mathrm{solve}(\mathbf{s}(x)=75, x=40)$ ▸ $x=49.3828$⚠

> **Obs**
> På grund af en fejl i programmet kan vi *ikke* anvende **nDerivative**-kommandoen, men må i stedet anvende den – i øvrigt mere præcise – **CentralDiff**-kommando.

Bemærkning: Sumkurven er netop stamfunktionen til tæthedsfunktionen (der omvendt netop er den afledede af sumkurven). Det kan vi kontrollere ved hjælp af de *numeriske* rutiner for differentiation og integration, idet forskrifterne jo *ikke* er opbygget af standardfunktioner. Vi skal da blot huske på, at vi har et ekstra skillepunkt i tæthedsfuntionen, fordi vi har inkluderet det første intervalendepunkt i tabellen, så indeks for tæthedsfunktionen skal løftes med én:

Tæthedsfunktionen:

$$f(x):=100\cdot\frac{\mathbf{hyppighed}\left[\mathrm{int}\left(\frac{x}{10}\right)+2\right]}{10.\cdot\mathrm{sum}(\mathbf{hyppighed})} \blacktriangleright \textit{Udført}$$

Sumkurve:

$$s(x):=\mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]+\frac{\mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+2\right]-\mathbf{frekvens}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]}{10}\cdot\left(x-\mathbf{skillepunkt}\left[\mathrm{int}\left(\frac{x}{10}\right)+1\right]\right) \blacktriangleright \textit{Udført}$$

$$f(45.) \blacktriangleright 1.30063$$

$$\mathrm{centralDiff}(s(x),x)|x=45. \blacktriangleright 1.30063$$

$$s(45) \blacktriangleright 69.2996$$

$$\int_0^{45.} f(x)\,dx \blacktriangleright 69.2996$$

6 Sandsynlighedsregning

TI-Nspire CAS stiller en lang række sandsynlighedsteoretiske funktioner til rådighed.

I dette afsnit skal du især se en enkelt af disse, tilfældig stikprøveudtagning, idet du skal på opdagelse i sandsynlighedsregningen ved at simulere tilfældige stikprøveudtagninger.

Simulering af terningkast

Obs
Tilfældig-tals generatoren skal have et tal at starte på (0 er standard). To computere vil derfor generere præcis de samme tilfældige tal, hvis de benytter samme starttal. Du kan bryde denne sammenhæng ved at *genberegne* regnearket mange gange ved at holde **Ctrl/Cmd R** -tasten nede.

Ved kast med en terning er der 6 *mulige udfald*: 1, 2, 3, 4, 5 og 6, der alle er lige sandsynlige. Her refererer tallet til antallet af øjne på den side af terningen, der vender opad. Men i det følgende er vi faktisk kun interesseret i om vi får en *sekser* eller en *ikke-sekser*. *Basiseksperimentet* består altså i et kast med en terning, og der er netop 2 mulige udfald: sekser (succes) eller ikke-sekser (fiasko), hvor sandsynligheden for at få en ikke-sekser er fem gang så stor som sandsynligheden for at få en sekser.

Du kan simulere kast med en terning ved at vælge et tilfældigt helt tal mellem 1 og 6. Hertil er TI-Nspire CAS udstyret med funktionen randSamp, der trækker en tilfældig stikprøve fra en population. Du kan tænke på populationen som et *sandsynlighedsfelt*. Populationen indeholder de mulige udfald, dvs. ”ikke-seksere” og ”seksere”. Men da der er fem gange så mange ”ikke-seksere” som ”seksere” på en terning, skal der også være fem gange så mange ”ikke-seksere” i populationen.

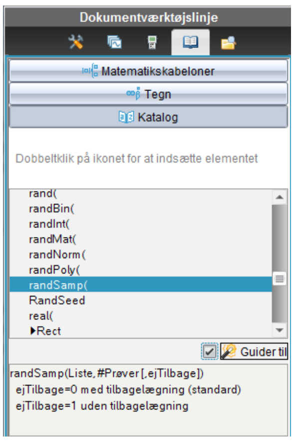
Vi går derfor ind i et **Lister og regneark**-værksted og opretter en søjle med titlen **population**, hvor vi indskriver værdierne ”ikke-sekser” fem gange (eller bare én gang, hvorefter værdien trækkes ned gennem de første fem celler) efterfulgt af en enkelt celle med værdien ”sekser”. Husk at tekstvariable altid skal i gåseøjne!

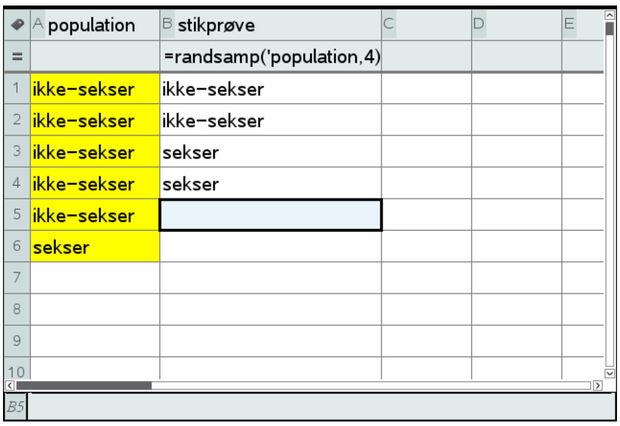
Hvad er nu *sandsynligheden* for at få (mindst) én sekser i 4 kast med en terning?

I den næste søjle, der navngives **stikprøve**, indskriver vi nu formlen

= RandSamp(**population**,4)

i det vi, som det sammensatte eksperiment, netop kaster 4 gange med terningen. Som udgangspunkt udtages stikprøven med *tilbagelægning*. Populationen er altså den samme hver eneste gang vi trækker et nyt element. Dermed er sandsynligheden for at få en sekser også den samme hver eneste gang vi trækker et nyt element. Vi trækker i alt fire elementer, fordi vi har fire forsøg til at få en sekser.





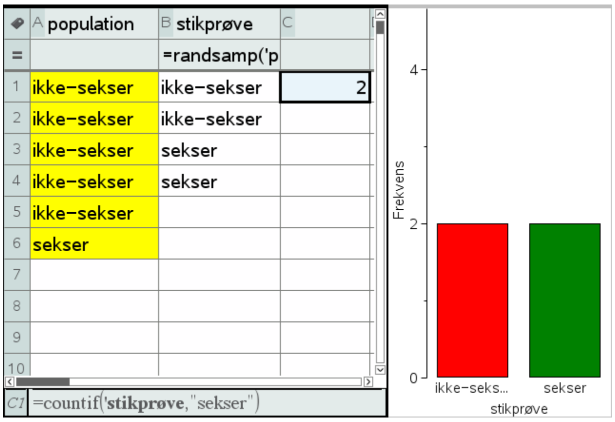
På skærmbilledet til højre er vist, hvordan du simulerer fire kast med én terning. Resultatet skal tolkes således, at du i første kast slår en ikke-sekser, i andet kast en ikke-sekser, i tredje kast en sekser og i fjerde kast en sekser. I denne omgang slog vi altså to seksere i fire forsøg.

For at finde svar på dette spørgsmål skal simulationen gentages mange gange, og undervejs skal der holdes regnskab med, om der kommer en sekser i et af de fire kast eller ej.

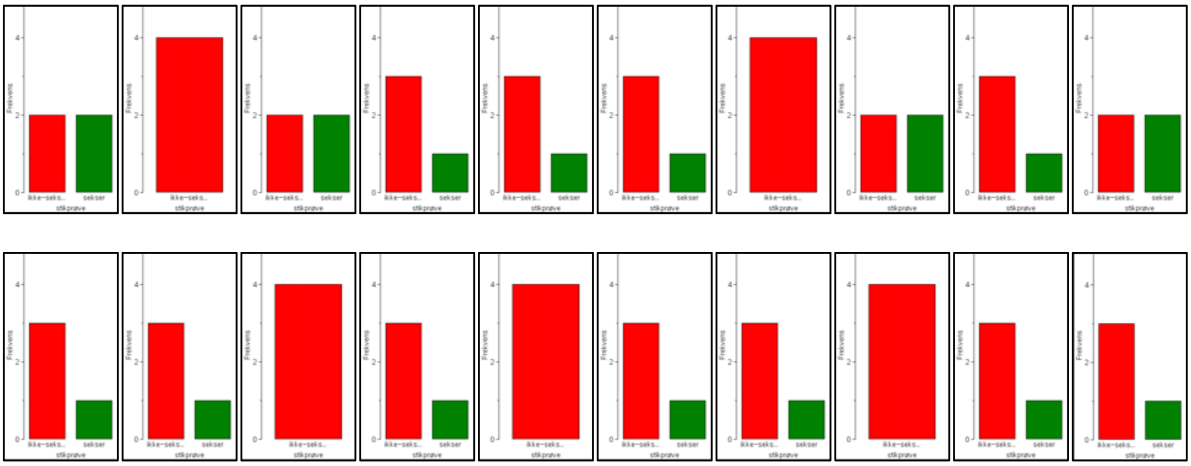
Vi opretter derfor først en celleformel i den næste søjle, der holder øje med antal seksere:

$$\boxed{C1} = \text{countif}\left(\textbf{stikprøve}, \text{"sekser"}\right)$$

Læg godt mærke til gåseøjne. Det er afgørende at du har indskrevet teksterne i populationslisten med gåseøjne, ligesom det er afgørende, at du har brugt gåseøjne i søgefeltet for countif-kommandoen.



I skærmbilledet ovenfor har vi også tilføjet et diagramvindue, så vi kan holde øje med om der kommer seksere (grøn søjle). Gentager du nu simuleringen fx 20 gange kan vi begynde at danne os et overblik over hvor nemt det er at få en sekser i fire forsøg med kast med en terning.



Vi ser da at der i løbet af de 20 simulationer var der 5 gange, hvor vi slet *ikke* fik en sekser (den grønne søjle mangler), der var 11 gange, hvor vi netop fik én sekser. Endelig var der 4 gange, hvor vi netop fik to seksere. En optælling (du får måske et andet resultat) viser altså, at i over halvdelen af de 20 gentagelser er der (mindst) én sekser.

Nu er 20 simuleringer selvfølgelig ikke så mange, så vi vil godt sætte antallet af simuleringer op til fx 1000. Samtidigt vil vi gerne registrere antallet af seksere i hver af de 1000 simuleringer. Det kan nu gøres på flere forskellige måder.

Datafangst med LUA-kommandoen *capture*

Den første metode vi viser benytter en særlig facilitet i **Lister og regneark**-værkstedet, en indbygget LUA-kommando *capture*. LUA er et særligt programmeringssprog, der bl.a. er meget udbredt ved programmering af computerspil. Men TI-Nspire CAS er faktisk også programmeret i LUA. Normalt er det ikke så synligt, men lige præcis med datafangst-kommandoen *capture*, bliver det faktisk afgørende, at det ikke er en sædvanlig matematik-kommando, men en LUA-kommando ☺

I computerspil har man typisk brug for at overvåge tilstanden hos helten, fx hvor stærk er vedkommende, hvilke magiske evner har vedkommende til rådighed, hvilke skatte har vedkommende samlet op undervejs osv. LUA kan derfor overvåge variable og se hvilke værdier de har. Der findes nu to typiske overvågningsmekanismer i computer spil programmeret i LUA, en *automatisk* og en *manuel*. I den *automatiske overvågningsmekanisme* registreres udelukkende om en variabel skifter værdi eller ej (hvilket så fx kan have konsekvenser for de udfordringer, som helten møder på sin vej). Men der er også en *manuel overvågningsmekanisme*, der registrerer værdien af variablen uanset om den ændrer sig eller ej, i samme øjeblik den manuelle overvågningsmekanisme udløses.

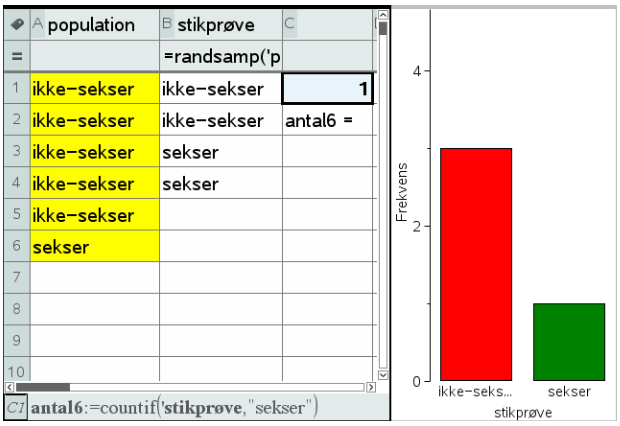
Her vil vi overvåge en variabel, der angiver antallet af seksere i en simulering, dvs. tallet registreret i celle C1, der var givet ved formlen

$$\boxed{C1} = \text{countif}(\textbf{stikprøve}, \text{"sekser"})$$

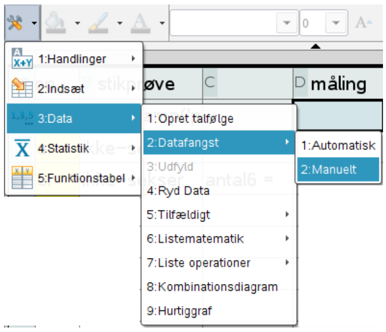
Vi skal da først have navngivet tallet som en variabel, så den er klar til overvågning. Det gør vi nemmest ved at tilføje navnet **antal6** foran lighedstegnet og samtidigt sætte et kolon lige foran lighedstegnet, som tegn på at vi tildeler værdien af variablen **antal6** det tal, der fremkommer ved udregningen af formlen countif(**stikprøve**,”sekser”). Celleformlen ser altså nu således ud:

$$\textbf{antal6} := \text{countif}(\textbf{stikprøve}, \text{"sekser"})$$

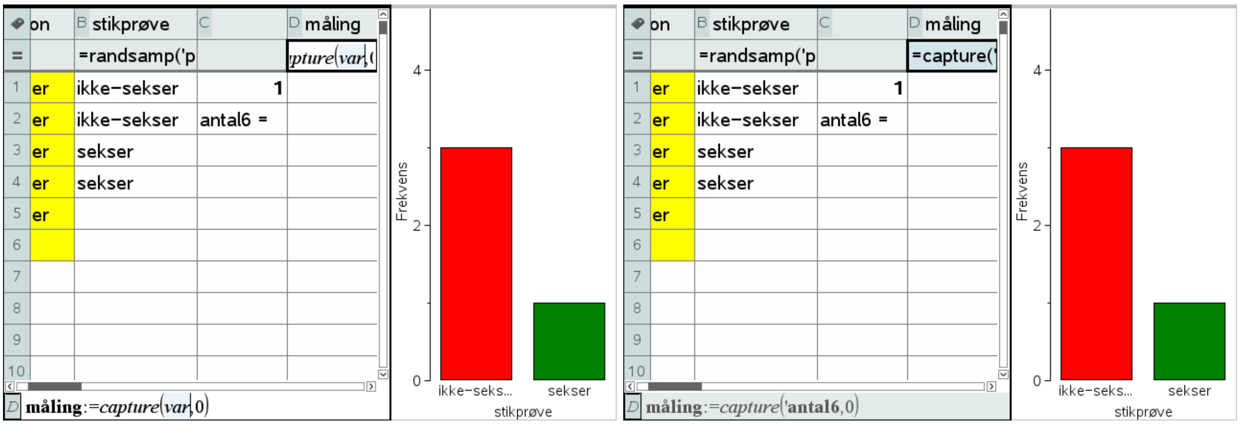
Læg mærke til at tallet nu skrives med **fedt** som tegn på, at det er lagret som en variabel. For at huske navnet har vi også skrevet navnet på variablen i cellen lige nedenunder ☺



Så er vi klar til at udføre en datafangst. Opret en ny søjle, kaldet **måling**, og vælg nu den følgende kommando i **Data**-menuen



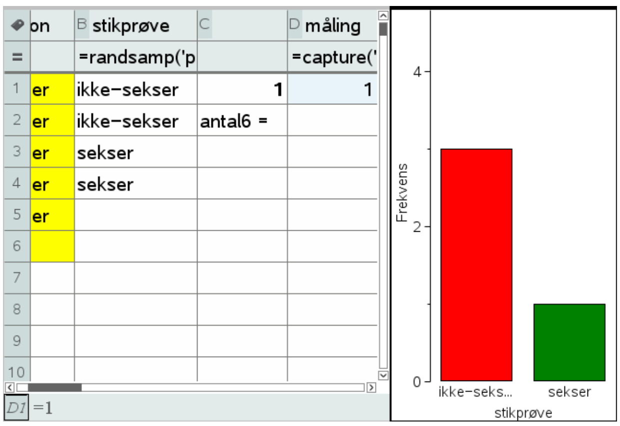
Her vælger vi en *manuel datafangst*, da antallet af seksere jo ikke nødvendigvis skifter værdi, bare fordi vi udfører en ny simulering ☺. Resultatet er den følgende formel:



Obs
Capture-kommandoen virker kun i **Lister og regneark**-værkstedet. Det er jo netop *ikke* en matematik-kommando, men en LUA-kommando. Derimod virker udløsningskommandoen **CTRL/CMD .** fra ethvert værksted.

Læg mærke til, at *capture*-kommandoen står i kursiv – det er netop *ikke* en matematik-kommando, men en LUA-kommando. Inde i denne *capture*-kommando skal du nu angive hvilken variabel vi skal fange, her variablen **antal6**. Endelig er der en parameter 0, der fortæller LUA-proceduren, at der er tale om en mauel datafangst.

Læg også mærke til, at der i første omgang slet ikke sker noget! Men det er jo fordi der er tale om en manuel datafangst, dvs. vi skal selv udløse datafangsten. Det sker ved at taste **CTRL/CDM .** (Kontrol Punktum). Det er ligegyldigt, hvor vi befinder os, når vi taster **CTRL/CMD .** – den manuelle datafangst udløses altid af denne taste-kombination, også selvom vi befinder os i et helt andet værksted.



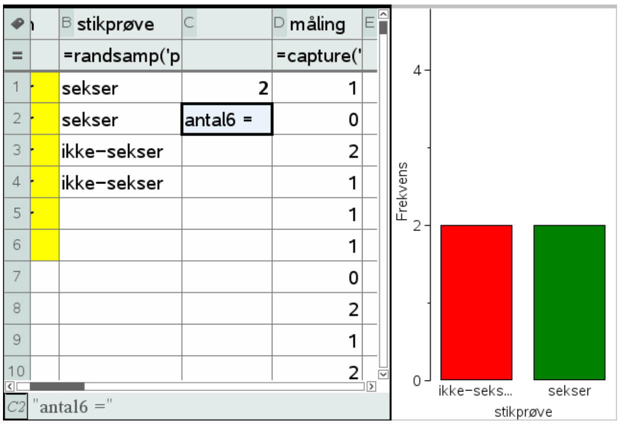
Nu skete der så endeligt noget! Vi har fanget antallet af seksere i den første simulering ☺

Obs
Husk at stå i **Lister og regneark**-værkstedet, for det er kun her at **CTRL/CMD R** udløser en genberegning af regnearket og dermed en ny simulering.

Så skal vi holde tungen lige i munden. Vi skal nu stille os i **Lister og regneark**-værkstedet og skiftevis taste **CTRL/CMD R** for at udføre en ny simulering henholdsvis **CTRL/CMD .** for at udløse en ny datafangst!

Det kan godt lyde lidt indviklet, men med lidt øvelse kan man fx holde **CTRL/CMD**-tasten ned med venstre hånd, mens man med højre hånd skiftevis taster **R** henholdsvis **Punktum**.

Resultatet af de første 10 simuleringer kan fx se således ud:

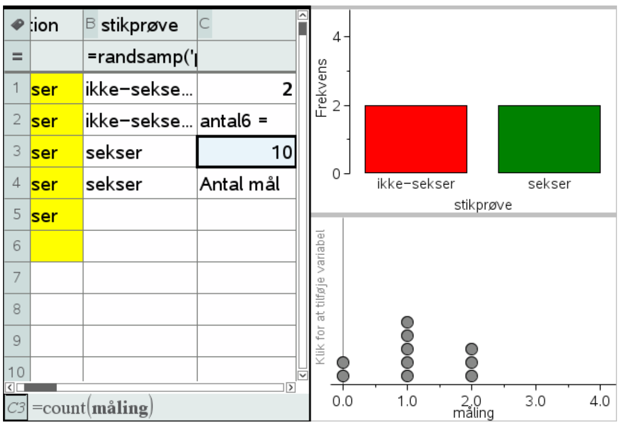


Inden vi går i gang med at lave datafangst i større omfang vil vi lige klargøre datafangsten, så vi dels kan se hvor mange målinger vi har foretaget, dels kan følge opbygningen af fordelingen for disse målinger.

Antallet af målinger fanger vi med celleformlen

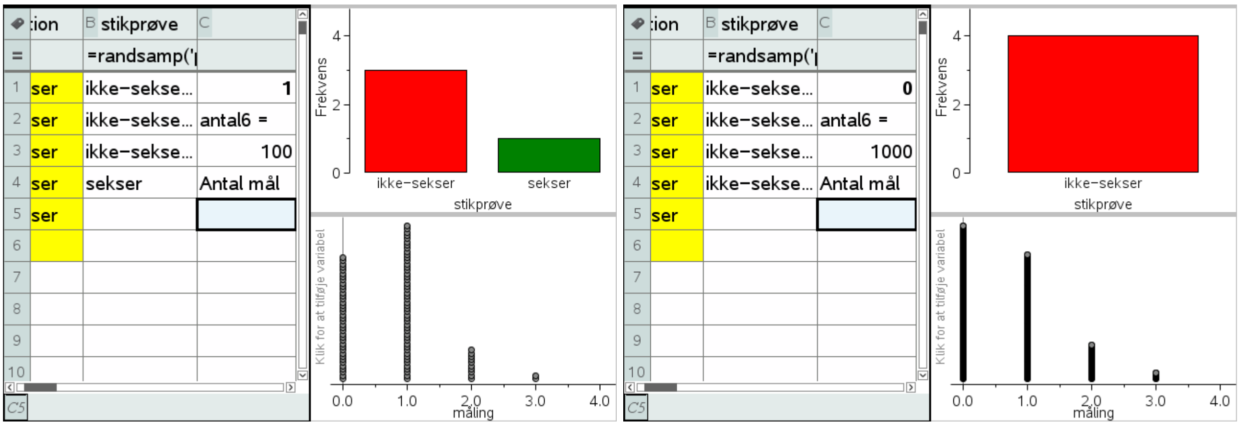
$$\boxed{C3} = \operatorname{count}(\mathbf{måling})$$

Visualiseringen af fordelingen klarer vi ved at oprette et nyt **Diagram og statistik**-vindue diagramtype til et prikdiagram for variablen **måling**. Her ses resultatet af første ti målinger:



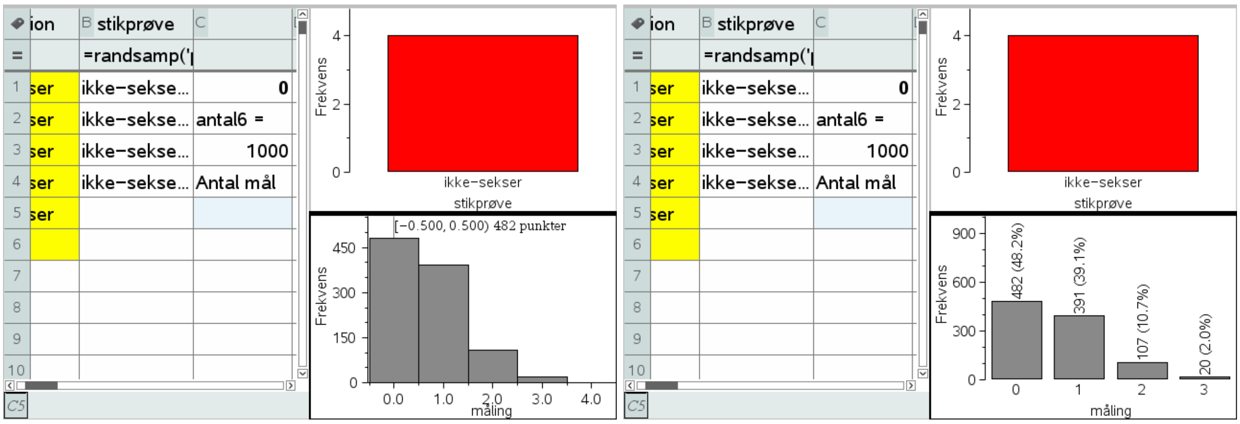
Vi er så klar til at danse med fingrene, idet vi først taster **CTRL/CMD R** og derefter taster **CTRL/CMD punktum**.

Vi viser først resultatet af de første 100 målinger, derefter resultatet af de første 1000 målinger.



Der tegner sig altså et mønster, hvor det hyppigste udfald er 0 seksere, men der er en hel del udfald med 2 seksere og faktisk også nogle få udfald med 3 seksere.

Vi kan danne os et bedre indtryk af fordelingen ved at skifte diagramtype til et histogram eller et søjlediagram (hvor vi så må tvinge målingen til at være en kategorisk variabel). I det sidste tilfælde kan vi så også slå **Vis alle etiketter** til:



I begge tilfælde ser vi da, at sandsynligheden for slet ikke at få en sekser er under 50%. Faktisk er vores skøn over sandsynligheden baseret på 1000 gentagne simuleringer givet ved 48.2%.

Sandsynligheden for at få mindst en sekser i fire forsøg synes altså at være over 50%!

Man kan selvfølgelig med rette synes det er bøvlet at gennemføre den ovenstående simulering 1000 gange.

Det ville da også være rart hvis man kunne flytte den over på en automatisk datafangst, der er meget nemmere at udføre i praksis. Vi har da det problem, at hvis målingen gentager sin værdi, registreres den *ikke* automatisk. Der findes snedige tricks til at omgå dette problem, men de er netop snedige og vi vil derfor ikke gennemgå dem i denne eksempelsamlingen. Men der findes fx et hæfte om **Statistik med TI-Nspire CAS**, hvor man kan finde detaljerne.

Avanceret datafangst med sekvens-kommandoer*

Vi vil nu se på mulighederne for at udføre datafangsten i ét hug, dvs. udregne antallet af seksere for 1000 simuleringer og opbygge en liste med de fundne værdier i ét hug, i stedet for trinvis, som vi så det med LUA-kommandoen *capture*. Hertil bruges sekvenskommandoen, der frembringer en talfølge ud fra et udtryk

$$\mathrm{seq}(\mathit{udtryk}, \mathit{indeks}, \text{startværdi}, \text{slutværdi})$$

Man kan sagtens arbejde med sekvenskommandoen seq i **Lister og regneark**, men ved mere komplicerede sekvenskommandoer er det mere overskueligt at flytte arbejdet over i et **Note**-værksted. Her viser vi først hvordan kommandoen virker ved at udregne en liste over de 10 første kvadrattal:

$$\mathrm{seq}(x^2, x, 1, 10) \blacktriangleright \{1,4,9,16,25,36,49,64,81,100\}$$

Men ellers går vi frem præcis ligesom før. Først oprettede vi en population, dernæst trak vi en stikprøve på fire elementer (med tilbagelæsning!) ved hjælp af RandSamp-kommandoen. Denne kommando er helt central i sekvenskommandoen, da det er den der styrer dynamikken og skal sikre, at vi får et nyt antal seksere i hver simulering. Endelig benyttede vi en Countif-kommando til at finde antallet af seksere i stikprøven. I første omgang tester vi nu vores sekvens-kommando ved blot at opsamle 10 værdier:

```
Først henter vi lige populationen:
population
 ▸ {"ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser","sekser"}
Så opstiller vi den formel, der udregner stikprøven:
randSamp(population,4) ▸ {"ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser"}
Så tæller vi hvor mange seksere, der var ved at bruge kommandoen
    countif(liste,"seksere"),
hvor liste erstattes af formlen for stikprøven:
countIf(randSamp(population,4),"sekser") ▸ 2
Endelig frembringer vi vi talfølgen ved hjælp af sekvenskommandoen
    seq(udtryk,x,1,10)
hvor udtryk erstattes af den ovenstående tælle−kommando
seq(countIf(randSamp(population,4),"sekser"),x,1,10) ▸ {1,1,1,0,2,1,2,1,1,0}
```

Som det kan ses virker **seq**-kommandoen! Hvis man indskriver den direkte i ét hug inde i fx et **Lister og regneark**-værksted, skal man være mordeligt forsigtigt med parenteserne, da det udtryk, der frembringer talfølgen, er sammensat af flere lag:

$$\mathrm{seq}\left(\boxed{\mathrm{countif}\left(\boxed{\mathrm{randSamp}(\mathbf{population},4)},\text{"sekser"}\right)},x,1,10\right)$$

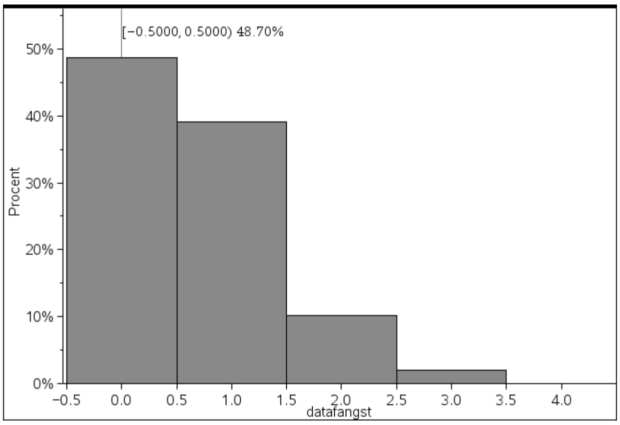
Det yderste lag består af tælle-kommandoen, mens det inderste lag består af stikprøve-kommandoen.

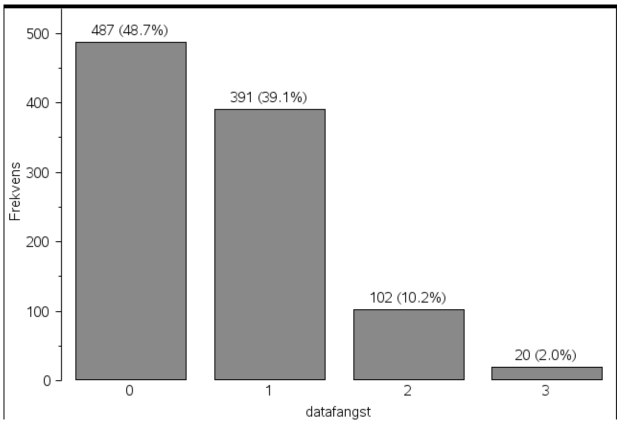
Men når først det virker kan vi jo roligt navngive listen og sætte antallet af gentagelser op til 1000:

```
Først henter vi lige populationen:
population
 ▸ {"ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser","sekser"}
Så opstiller vi den formel, der udregner stikprøven:
randSamp(population,4) ▸ {"ikke−sekser","ikke−sekser","ikke−sekser","ikke−sekser"}
Så tæller vi hvor mange seksere, der var ved at bruge kommandoen
    countif(liste,"seksere"),
hvor liste erstattes af formlen for stikprøven:
countIf(randSamp(population,4),"sekser") ▸ 2
Endelig frembringer vi vi talfølgen ved hjælp af sekvenskommandoen
    seq(udtryk,x,1,10)
hvor udtryk erstattes af den ovenstående tælle−kommando
seq(countIf(randSamp(population,4),"sekser"),x,1,10) ▸ {1,1,1,0,2,1,2,1,1,0}
datafangst:=seq(countIf(randSamp(population,4),"sekser"),x,1,1000)
 ▸ {1,0,1,1,1,0,1,1,0,1,0,1,0,0,1,1,0,1,0,0,0,1,0,0,1,2,0,1,1,1,0,1,1,0,0,0,0,0,0,1,1,1,1,0,2,1,1,1,1,
```

6: Sandsynlighedregning

Der går et splitsekund og vi har hentet de 1000 målinger! For at kunne overskue dem kan vi derefter afbilde dem i et **Diagrammer og Statistik**-værksted, hvor vi enten kan afbilde dem som et *histogram* (numerisk variabel) eller som et *søjlediagram* (kategorisk variabel)





I begge tilfælde ser vi da igen, at sandsynligheden for slet ikke at få en sekser er under 50%. Denne gang er vores skøn over sandsynligheden baseret på 1000 gentagne simuleringer givet ved 48.7%.

Sandsynligheden for at få mindst en sekser i fire forsøg (der jo er summen af de resterende sandsynligheder) synes altså stadig at være over 50%!

Man kan synes det er lidt kompliceret med de indpakkede sekvenskommandoer. Hvis man kender lidt til programmering er det da også muligt at oprette en brugerdefineret funktion, med en langt mere overskuelig opbygning. Men i dette hæfte vil vi ikke komme nærmere ind på programmering med TI-Nspire CAS. Det kan man i stedet læse mere om i et særskilt hæfte om **Programmering med TI-Nspire CAS**.

En simpel sansynlighedsberegning

Som vi har set får vi lidt forskellige skøn over sandsynligheden for slet ingen seksere, når vi simulerer 1000 gange – i det ene tilfælde fandt vi 48.2%, i det andet tilfælde 48.7%.

Men lige præcis denne sandsynlighed er så simpel at beregne, at vi nemt kan finde den eksakte værdi. Sandsynligheden for ikke at få en sekser i et enkelt kast med en terning er selvsagt 5/6. Sandsynligheden for slet ikke at få seksere i fire forsøg i træk er derfor produktet af denne sandsynlighed fire gange med sig selv, dvs. vi finder sandsynligheden

$$\left(\frac{5}{6}\right)^4 = \frac{625}{1296} \qquad \left(\frac{5.}{6.}\right)^4 \approx 0.482253$$

Den eksakte sandsynlighed er altså 625/1296, som er klart mindre end en halv, idet $2 \cdot 625 = 1250$. Udtrykt som decimaltal finder vi altså også at sandsynligheden med 2 decimaler er givet ved

48.23%.

Det er altså en klar fordel at satse på at få seksere, hvis man får lov til at kaste fire gange med en terning.

Chevalier de Meres problem

Omkring 1650 led Chevalier de Mere svære økonomiske tab som følge af, at han havde ræsonneret sig frem til, at følgende to spil har samme gevinstchance:

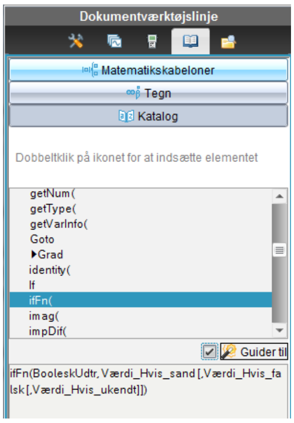
Spil 1: Kast en terning 4 gange og indgå et væddemål om at få en 6'er.
Spil 2: Kast to terninger 24 gange, og indgå et væddemål om at få en dobbelt-6'er

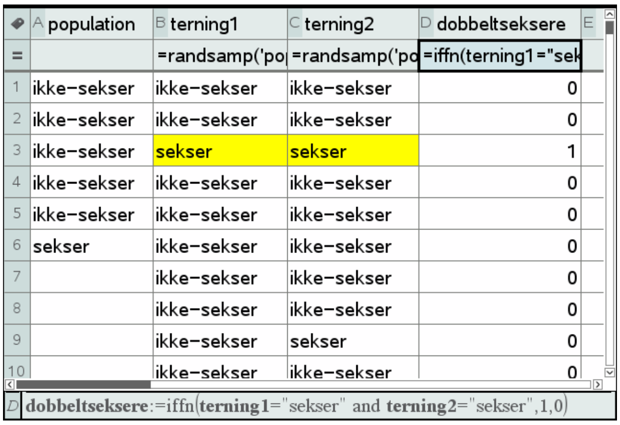
Et simpelt proportionalitetsargument, *som desværre ikke holder*, kunne fx være: I spil 2 gælder det om at få en dobbelt sekser med kast med to terninger. Det er seks gange så svært, som et få en enkelt sekser med kast med én terning. Derfor skal vi også have seks gange så mange forsøg, altså 24 forsøg med kast med to terninger.

Naivt kunne man altså forvente samme sandsynlighed for at vinde i de to spil.

Undersøgelsen i sidste afsnit viser, at sandsynligheden for at spil1 lykkes er omkring 51.77% – og altså større end 50%. Med andre ord hævder de Mere, at sandsynligheden for, at det lykkes, er større end 0.5 i begge spil. Men gælder det nu også for det andet spil?

Her kan vi jo prøve at simulere os frem til et svar. Der er mange måder at udføre simuleringen på. Her viser vi en, der ligger tæt op af det faktiske eksperiment: 24 kast med to terninger.





	A population	B terning1	C terning2	D dobbeltseksere	E
=		=randsamp('po	=randsamp('po	=iffn(terning1="sek	
1	ikke−sekser	ikke−sekser	ikke−sekser	0	
2	ikke−sekser	ikke−sekser	ikke−sekser	0	
3	ikke−sekser	sekser	sekser	1	
4	ikke−sekser	ikke−sekser	ikke−sekser	0	
5	ikke−sekser	ikke−sekser	ikke−sekser	0	
6	sekser	ikke−sekser	ikke−sekser	0	
7		ikke−sekser	ikke−sekser	0	
8		ikke−sekser	ikke−sekser	0	
9		ikke−sekser	sekser	0	
10		ikke−sekser	ikke−sekser	0	

D **dobbeltseksere**:=iffn(**terning1**="sekser" and **terning2**="sekser",1,0)

Her er der først oprettet en population med fem "ikke-seksere" og en "sekser". Derefter to stikprøver terning1 og terning2 med kommandoen randSamp(**population**,24), dvs. vi får lov til at kaste 24 gange med begge terninger. Vi skal så se efter dobbeltseksere. Det gør vi med en ny søjle, kaldet **dobbeltseksere**, der udnytter en betinget kommando ifFn (dvs. if-funktionen – der findes også en **if**-kommando, men den er reserveret programmering).

Den tjekker altså om de to terningekast samtidigt giver seksere. Hvis det er tilfældet har den værdien 1 (en gevinst) og ellers værdien 0 (en nitte). I det ovenstående tilfælde gav det tredje kast med to terninger altså anledning til en dobbeltssekser.

Vi kan så nemt finde det samlede antal dobbeltseksere ved hjælp af celleformlen

$$\boxed{E1} = \text{sum}(\textbf{dobbeltseksere})$$

I dette tilfælde var der altså to dobbeltseksere, idet det viste sig, at det syttende kast også gave en dobbeltsekser.

	B terning1	C terning2	D dobbeltseksere	E	F
=	=randsamp('po	=randsamp('po	=iffn(terning1="sek		
1	ikke–sekser	ikke–sekser	0	2	
2	ikke–sekser	ikke–sekser	0	antal66 =	
3	sekser	sekser	1		
4	ikke–sekser	ikke–sekser	0		
5	ikke–sekser	ikke–sekser	0		
6	ikke–sekser	ikke–sekser	0		
7	ikke–sekser	ikke–sekser	0		
8	ikke–sekser	ikke–sekser	0		
9	ikke–sekser	sekser	0		
10	ikke–sekser	ikke–sekser	0		

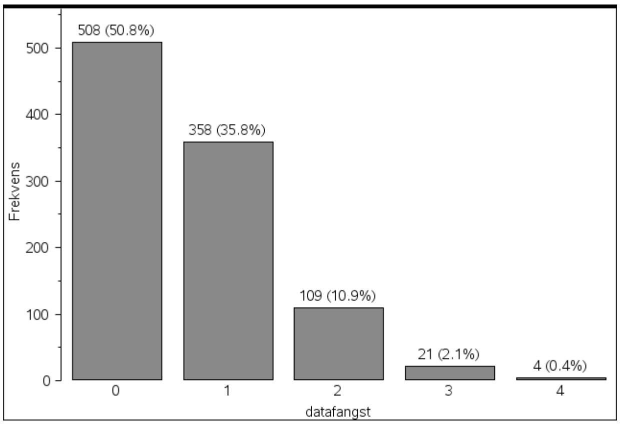
E1 =sum('dobbeltseksere)

	B terning1	C terning2	D dobbeltseksere	E	F
=	=randsamp('po	=randsamp('po	=iffn(terning1="sek		
16	ikke–sekser	ikke–sekser	0		
17	sekser	sekser	1		
18	ikke–sekser	sekser	0		
19	ikke–sekser	ikke–sekser	0		
20	ikke–sekser	sekser	0		
21	sekser	ikke–sekser	0		
22	ikke–sekser	ikke–sekser	0		
23	ikke–sekser	ikke–sekser	0		
24	ikke–sekser	ikke–sekser	0		
25					

B17:C17 ="sekser"

Du kan nu udføre 1000 simuleringer ligeom vi gjorde med det første spil, ved enten at benytte LUA-kommandoen *capture* eller udnytte seq-kommandoen. Det sidste viser vi i detaljer med de næste to skærmbilleder

```
population
 ▸ {"ikke–sekser","ikke–sekser","ikke–sekser","ikke–sekser","ikke–sekser","sekser"}
Så opstiller vi den formel, der udregner stikprøven for den ene terning:
randSamp(population,24)
 ▸ {"sekser","ikke–sekser","sekser","ikke–sekser","ikke–sekser","ikke–sekser","ikke–sekser'
Så tæller vi hvor mange dobbeltseksere, der var ved at bruge kommandoen
    sum(iffn(liste1 = "sekser" and liste2 = "sekser",1,0)),
hvor liste1 og liste2 erstattes af formlen for den ene henholdsvis den anden terning:
sum(ifFn(randSamp(population,24)="sekser" and randSamp(population,24)="sekser",1,0)) ▸ 3
Endelig frembringer vi vi talfølgen ved hjælp af sekvenskommandoen seq(udtryk,x,1,10)
hvor udtryk erstattes af den ovenstående tælle–kommando
seq(sum(ifFn(randSamp(population,24)="sekser" and randSamp(population,24)="sekser",1,0)),
x,1,10)
 ▸ {1,0,3,3,0,2,3,1,1,0}
datafangst:=seq(sum(ifFn(randSamp(population,24)="sekser" and randSamp(population,
24)="sekser",1,0)),x,1,1000)
 ▸ {1,1,0,2,1,2,0,0,0,0,1,3,0,0,0,0,0,1,0,1,1,0,0,0,1,1,0,0,0,1,1,2,0,1,1,0,0,4,0,0,1,1,0,0,0,0,0,0,0
```



Denne gang ser sandsynligheden ud til at få mindst en dobbeltsekser altså ud til at være mindre end 0.5.

Igen kan vi underbygge det eksperimentelle resultat med en simpel sandsynlighedsberegning. Sandsynligheden for ikke at få en dobbeltsekser i et enkelt kast med to terninger er 35/36. Sandsynligheden for slet ikke at få dobbeltseksere i fire og tyve forsøg i træk er derfor produktet af denne sandsynlighed fire og tyve gange med sig selv, dvs. vi finder sandsynligheden

$$\left(\frac{35}{36}\right)^{24} = \frac{114191312420705803871750831604003906\,25}{224522577073545572400872111237926748\,16} \qquad \left(\frac{35.}{36.}\right)^{24} \approx 0.508596$$

Den eksakte sandsynlighed er altså 114…/224…, som er klart større end en halv, idet $2 \cdot 114... = 228...$. Udtrykt som decimaltal finder vi også at sandsynligheden med 2 decimaler er givet ved

50.86%.

Det er altså en klar fordel *ikke* at satse på at få dobbeltseksere, hvis man får lov til at kaste fire og tyve gange med to terninger. At denne sandsynlighed er større end 50% lærte Chevalier de Mere på den hårde måde via simuleringer på casinoer. Flere kilder hævder, at han blev så godt som ruineret på dette spil. Han forelagde problemet for en af datidens klogeste hoveder, Blaise Pascal (1623 - 1662), som naturligvis løste det og lagde grundstenen til moderne sandsynlighedsregning.

Binomialfordelingen

Binomialfordelingen bygger på *et sammensat eksperiment*, hvor man gentager et *basiseksperiment n* gange, hvor *n* kaldes *antalsparameteren*. Basiseksperimentet har netop to mulige udfald, som vi i almindelighed betegner ”succes” og ”fiasko”, men i praksis kan vi selvfølgelig anvende mere præcise navne, som vi fx gjorde med ”sekser” og ”ikke-sekser” i forbindelse med terningespillene. Sandsynligheden for at få succes i basiseksperimentet betegnes *p* og kaldes *sandsynlighedsparameteren*. *Antallet af succeser X* i det sammensatte eksperiment er da binomialfordelt med antalsparameter *n* og sandsynlighedsparameter *p*.

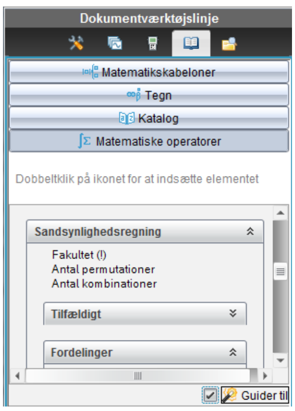
Vi har allerede set på hvordan man kan simulere en binomialfordeling:

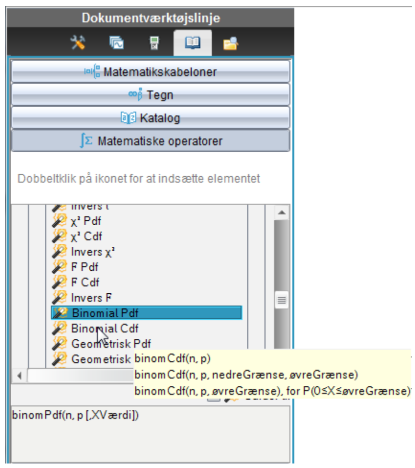
Først oprettes en **population** bestående af et passende antal ”succes” og ”fiasko”, hvor de indbyrdes forhold mellem antal ”succes” og antal ”fiasko”, netop afspejler de indbyrdes forhold mellem sandsynlighederne. Når vi trækker et element fra populationen skal der altså netop være sandsynligheden *p* for at trække en ”succes”. Basiseksperimentet består da netop i at trække et enkelt element fra populationen.

Derefter oprettes en **tilfældig stikprøve**, hvor vi trækker *n* elementer fra populationen *med tilbagelægning*, så populationen er uændret efter hver udtrækning. Udtagningen af den tilfældige stikprøve svarer da netop til det sammensatte eksperiment.

Endelig tæller vi op **hvor mange succeser** *X*, der er i den tilfældige stikprøve. Det er antallet af succeser, der er binomialfordelt. Den stokastisk binomialfordelte variabel *X* kan da antage værdierne fra 0 til *n*. I den matematiske teori for binomialfordelingen udleder man nu en formel for *sandsynlighedsfordelingen*, dvs. sandsynlighederne for de enkelte værdier af *X*.

Ikke overraskende er binomialfordelingen indbygget i TI-Nspire CAS. Slår man op i fortegnelsen over **Matematiske operatorer** under **Sandsynlighedsregning** og **Fordelinger** finder man netop to fordelinger knyttet til binomialfordelingen, den såkaldte punktfordeling (Point Distribution Function) **binomPdf** og den såkaldte intervalfordeling (Cumulative Distribution Function) **binomCdf**. Det er altså P’et henholdsvis C’et der adskiller de to navne ☺





Behersker man disse to fordelingsfunktioner kan man nemt løse alle praktiske opgaver med binomialfordelinger. Vi ser derfor på dem i lidt større deltalje:

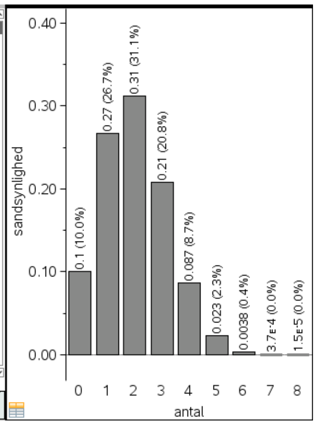
Punktfordelingen: BinomPdf(*n*,*p*,[*x*])
Den kræver – ikke overraskende – at man først oplyser de to parametre: antalsparameteren *n* og sandsynlighedsparameteren *p*. Lidt mere overraskende er det nok, at *x*-værdien (antal succeser) er valgfri, men droppes den fås simpelthen hele sandsynlighedsfordelingen som

en liste. Det er et elegant træk, der gør det nemt at oprette tabeller over binomialfordelinger i regneark, ligesom det er nemt at tilknytte grafer til fordelingen.

Her ses for eksempel sandsynlighedsfordelingen for binomialfordelingen med $n = 8$ og $p = 1/4$ både som tabel og graf

	A antal	B sandsynlighed
=		=binompdf(8,1/4)
1	0	0.100113
2	1	0.266968
3	2	0.311462
4	3	0.207642
5	4	0.086517
6	5	0.023071
7	6	0.003845
8	7	0.000366
9	8	0.000015
10		

A:B 0



Men punktfordelingen kan selvfølgelig også udnyttes til konkrete udregninger, som fx

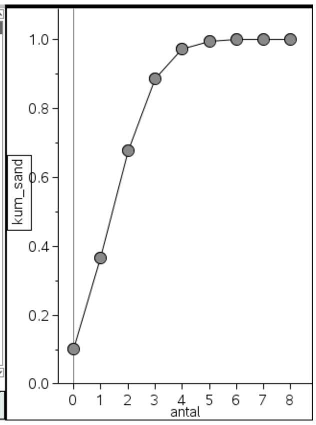
> Vi ønsker at finde sandsynligheden $p(X=2)$, hvor X er binomialfordelt med antalsparameter $n = 8$ og sandsynlighedsparameter $p = 1/4$.
>
> $\mathrm{binomPdf}\left(8,\frac{1}{4},2\right) \blacktriangleright 0.311462$
>
> Den søgte sandsynlighed for netop to succeser er altså 31.15%.

Intervalfordelingen: BinomCdf(n,p,[a],[b])
Igen skal man først oplyse de to parametre: antalsparameteren n og sandsynlighedsparameteren p. Lidt mere overraskende er det nok, at intervalgrænserne a og b er valgfri, men droppes de fås simpelthen hele den kumulerede sandsynlighedsfordeling som en liste. Det er et elegant træk, der gør det nemt at oprette tabeller over kumulerede binomialfordelinger i regneark, ligesom det er nemt at tilknytte grafer til fordelingen.

Her ses for eksempel den kumulerede sandsynlighedsfordeling for binomialfordelingen med $n = 8$ og $p = 1/4$ både som tabel og graf

	A antal	B kum_sand
=		=binomcdf(8,
1	0	0.100113
2	1	0.367081
3	2	0.678543
4	3	0.886185
5	4	0.972702
6	5	0.995773
7	6	0.999619
8	7	0.999985
9	8	1.
10		

B1 =0.10011291503906



Men intervalfordelingen kan selvfølgelig også udnyttes til konkrete udregninger, som fx

Vi ønsker at finde sandsynligheden $p(2 \le X \le 5)$, hvor X er binomialfordelt med antalsparameter $n = 8$ og sandsynlighedsparameter $p = 1/4$.

$$\text{binomCdf}\left(8, \frac{1}{4}, 2, 5\right) \blacktriangleright 0.628693$$

Den søgte sandsynlighed for at antallet af succeser er mindst 2 og højst 5 er altså 62.87%.

Til slut lægger vi mærke til at intervalfordelingen strengt taget er overflødig, når vi har punktfordelingen, idet der jo fx gælder

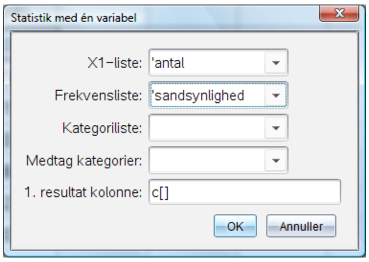
$$p(2 \le X \le 5) = p(2) + p(3) + p(4) + p(5)$$

Vi kunne altså ligeså godt have beregnet intervalsandsynligheden som en sum over de involverede punktsandsynligheder:

$$\sum_{x=2}^{5} \left(\text{binomPdf}\left(8, \frac{1}{4}, x\right)\right) \blacktriangleright 0.628693$$

Men det kræver selvfølgelig en vis fortrolighed med sum-tegnet ☺

Når vi kan styre punktfordelingen kan vi også udføre simple statistiske beregninger på sandsynlighedsfordelingen. Vi skal da bare opfatte sandsynlighedsfordelingen som en *vægtet population*, dvs. de enkelte udfald fra 0 til 8 optræder med en vægt, der netop er angivet ved sandsynligheden: Vi udfører derfor en enkeltvariabelstatistik med **sandsynlighed** som frekvensliste



	antal	B sandsynlighed	C	D
=		=binompdf(8,1/4)		=OneVar('antal,'sandsynli
1	0	0.100113	Titel	Statistik med én variabel
2	1	0.266968	x̄	2.
3	2	0.311462	Σx	2.
4	3	0.207642	Σx²	5.5
5	4	0.086517	sx := sn-1x	#UNDEF
6	5	0.023071	σx := σnx	1.22474
7	6	0.003845	n	1.
8	7	0.000366	MinX	0.
9	8	0.000015	Q1X	1.
10			MedianX	2.

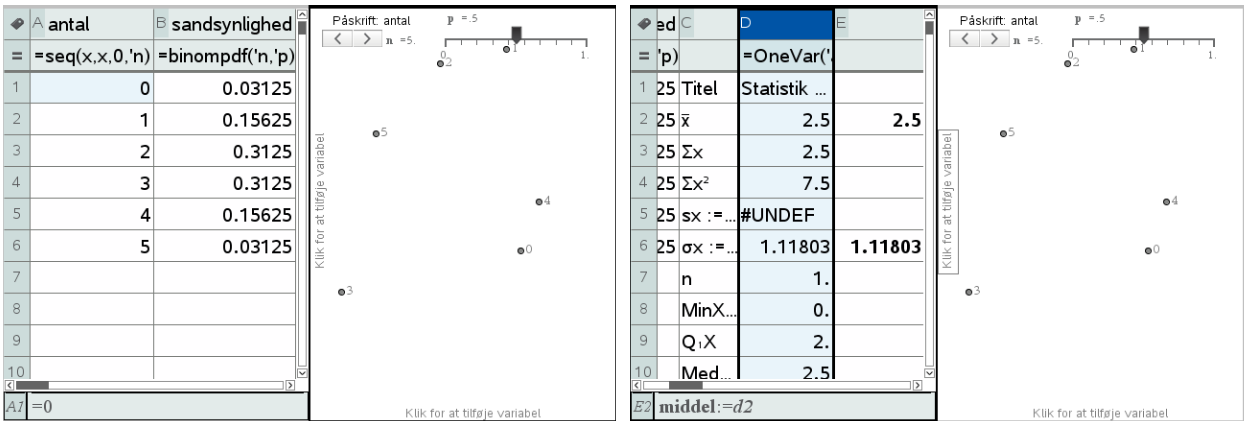
D1 ="Statistik med én variabel"

De vigtigste oplysninger er da middelværdien $\bar{x}$ som er 2, og spredningen σx som har værdien 1.22474. Nederst i listen finder man faktisk også variansen SSX (dvs. kvadratet på spredningen)

SSX := Σ(x−x̄)²	1.5

Variansen har altså værdien 1.5 .

Det åbner muligheden for eksperimentelt at finde nogle af de formler, der gælder for middelværdi, spredning og varians. Vi kan nemlig indføre skydervariable for antalsparameteren n og sandsynlighedsparameteren p, og så udføre en dataopsamling, der viser de relevante sammenhænge. Det er nemmest at undersøge variabelsammenhængene for middelværdien og variansen. Vi starter med at oprette den dynamiske sandsynlighedsfordeling for binomialfordelingen efterfulgt af en enkeltvariabel statistik over de vægtede antal, samt indfører en minimeret 'diskret' skyder for n, der vokser i trin af størrelsen 1 fra 2 til 50, samt en 'kontinuert' skyder for p, der fx vokser i trin af 0.01:



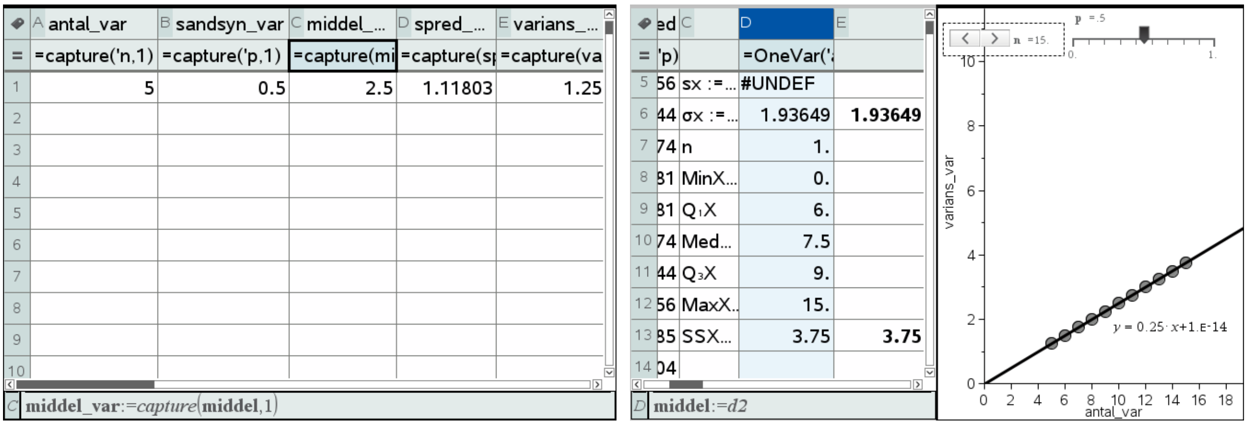
Vi trækker så værdierne for middelværdi, spredning og varians ud via celleformler i nabocellerne, så vi kan lave datafangst på dem. Her er det vist for middelværdien i celle E2, der er lagret som variablen middel via celleformlen

$$\boxed{E2}\ \mathbf{middel} := d2$$

Tilsvarende er spedningen, og variansen gemt som variablene

$$\boxed{E6}\ \mathbf{spredning} := d6 \qquad \boxed{E13}\ \mathbf{varians} := d13$$

Vi opretter så et nyt lister og regneark, hvor vi fanger værdierne for antalsparameteren *n*, sandsynlighedsparameteren *p*, middelværdien, spredningen og variansen

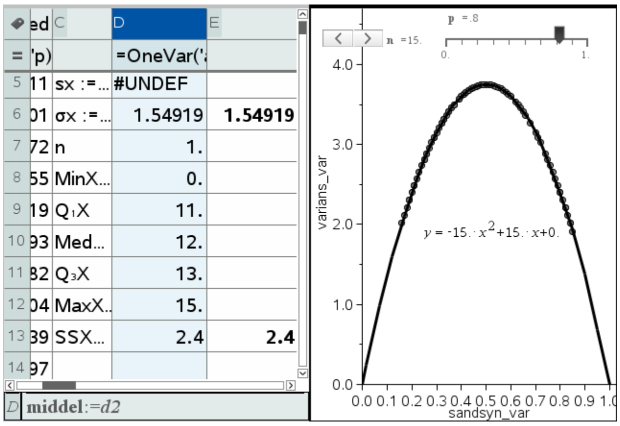


Så er vi klar til datafangsten ☺. Vi skal udføre variabelkontrol, hvor vi somme tider varierer *n* og sommetider *p*, men ikke dem begge på én gang. Det er nemmest at udføre disse nærtbeslægtede eksperimenter, hvis du opretter en række kopier af dette dokument, så du kan variere de uafhængige variable *n* og *p* systematisk og opbygge grafer for de afhængige variable **middel** og **varians**. Her ser vi på to typiske deleksperimenter: Hvis vi vil undersøge hvordan variansen afhænger af antalsparameteren, skal vi variere antalsparameteren og oprette et diagram, der viser variansen som funktion af antalsparameteren. Vi ser da tydeligt at variansen er direkte proportional med antalsparameteren *n*, dvs. der gælder en formel af typen

$$\text{varians} = k_1(p)\cdot n$$

hvor proportionalitetsfaktoren k_1, i vores tilfælde 0.25, stadigvæk kan afhænge af sandsynlighedsparameteren *p*, som vi jo har holdt konstant (variabelkontrol!).

Vi gentager så eksperimentet, men denne gang varierer vi sandsynlighedsparameteren *p*. Vi skal da først nulstille datafangsten, enten ved at arbejde i et nyt dokument, eller ved i det gamle dokument at gå ind og gentage alle capturekommandoerne (taste **ENTER** i formelfelterne) ☺



Denne gang er der tydeligvis tale om et andengradspolynomium, der går gennem 0 og 1 på førsteaksen (p-aksen). Denne gang er variansen altså givet ved en formel af typen:

$$\text{varians} = k_2(n) \cdot p \cdot (1-p)$$

hvor proportionalitetsfaktoren k_2, i vores tilfælde 15, stadigvæk kan afhænge af antalsparameteren n (der netop har værdien 15 ☺), som vi jo har holdt konstant (variabelkontrol!).

Tilsammen antyder eksperimentet altså kraftigt, at der gælder en sammenhæng af formen

$$Varians = k \cdot n \cdot p \cdot (1-p)$$

Her er proportionaliteskonstanten i vores tilfælde endda ganske enkelt 1, så vi har udledt den følgende variansformel eksperimentelt

$$Varians = n \cdot p \cdot (1-p)$$

Den tilhørende spredningsformel må da være givet ved

$$\sigma = \sqrt{n \cdot p \cdot (1-p)}$$

Prøv nu selv at lege med middelværdien for en binomialfordeling. Her er resultatet knap så overraskende: Hvis vi har fx 8 forsøg og vi i hvert forsøg har sandsynligheden 0.25 for at få succes, så forventer vi i middel at fange $8 \cdot 0.25 = 2$ succeser, dvs. middelværdiformlen i almindelighed må være givet ved

$$\mu = n \cdot p$$

Desværre findes der ikke tilsvarende simple argumenter for spredningsformlen.

Gensyn med Chevalier de Meres problem

Nu hvor vi kan styre binomialfordelingerne kan vi se nærmere på sandsynlighedsfordelingerne i Chevalier de Meres to spil:

Spil 1: Kast en terning 4 gange og indgå et væddemål om at få en 6'er.
Spil 2: Kast to terninger 24 gange, og indgå et væddemål om at få en dobbelt-6'er

Overvej nu først:

Spil 1: Hvis X betegner antallet af 6'ere i 4 kast med en terning, så er X binomialfordelt med parametrene $n = 4$ og $p = \frac{1}{6}$

Spil 2: Hvis X betegner antallet af dobbelt-6'ere i 24 kast med to terninger, så er X binomialfordelt med parametrene $n = 24$ og $p = \frac{1}{36}$

Læg mærke til, at middelværdien i begge spillene er den samme:

$$\mu_1 = 4 \cdot \frac{1}{6} = \frac{2}{3}, \quad \mu_2 = 24 \cdot \frac{1}{36} = \frac{2}{3}$$

Det er jo netop på grund af Chevalier de Meres omhyggeligt konstruerede proportionalitet mellem de to spil.

Men spredningerne er *ikke* helt ens, idet vi finder

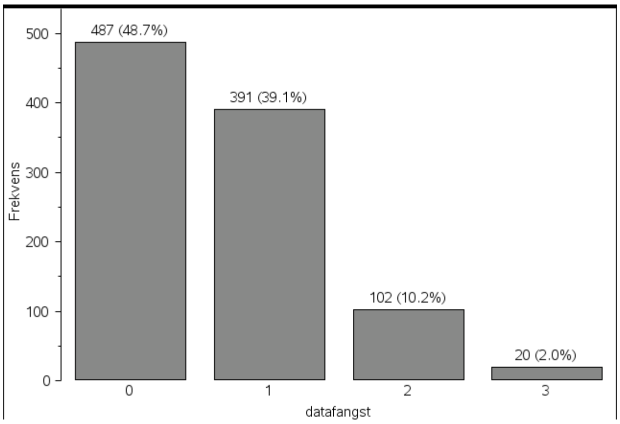
$$\sigma_1 = \sqrt{n_1 \cdot p_1 \cdot (1-p_1)} = \sqrt{4 \cdot \frac{1}{6} \cdot \frac{5}{6}} = \sqrt{\frac{2}{3} \cdot \frac{5}{6}} = \sqrt{\frac{2}{3} \cdot \frac{30}{36}} \approx 0.745$$

$$\sigma_2 = \sqrt{n_2 \cdot p_2 \cdot (1-p_2)} = \sqrt{24 \cdot \frac{1}{36} \cdot \frac{35}{36}} = \sqrt{\frac{2}{3} \cdot \frac{35}{36}} \approx 0.805$$

Spredningen i det første spil er altså tydeligvis lidt mindre end spredningen i det andet spil: Det har afgørende konsekvenser for de to spil! Da spredningen for spil 2 er større end spredningen for spil 1 og de jo har samme middelværdi 2/3 bliver der også tildelt mere sandsynlighed til $X_2 = 0$ og $X_2 = 4$ i spil 2. Det er især det første, der er fatalt, for det betyder i praksis at $p(X_2 = 0)$ ryger op over 50% og at det derfor slet ikke kan betale sig at deltage i spil 2. Men det betyder også, at når vi fx udfører 1000 simuleringer af spil 2, så ser vi rent faktisk udfald med $X_2 = 4$, hvad vi ikke gjorde i spil 1 (selv om det nu ikke er helt udelukket, så du kan have en anden serie af 1000 simuleringer, hvor der rent faktisk krøb et udfald med $X_2 = 4$ ind i søjlediagrammet) ☺.

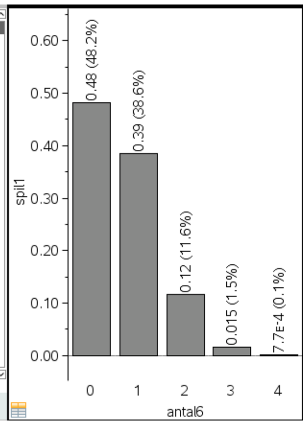
Her er relevante grafer (hvor de simulerede grafer er hentet fra et tidligere afsnit).

Først **spil 1** (hvor sandsynligheden for $X_1 = 0$ ligger under 50%, mens sandsynligheden for $X_1 = 4$ ligger under 1 ‰):



	A antal6	B spil1
=	=seq(x,x,0,4)	=binompdf(4,1/6)
1	0	0.482253
2	1	0.385802
3	2	0.115741
4	3	0.015432
5	4	0.000772
6		
7		
8		
9		
10		

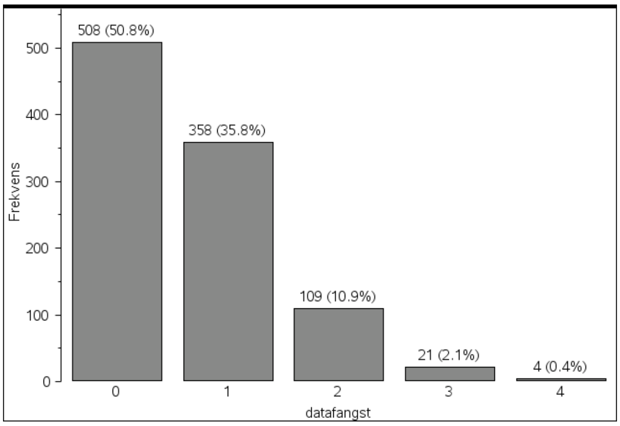
A antal6:=seq(x,x,0,4)



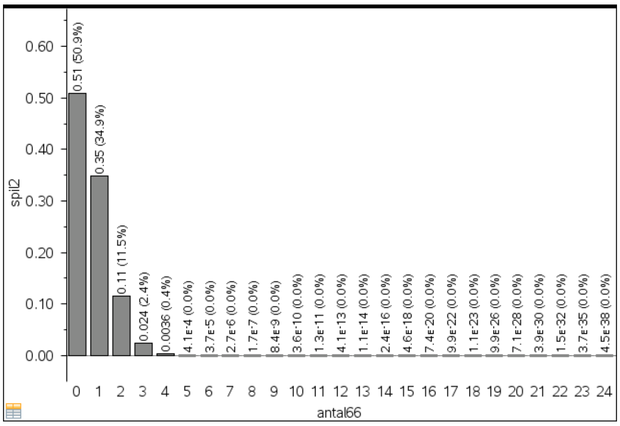
1000 simuleringer

Teoretisk fordeling

Derefter **spil 2** (hvor sandsynligheden for $X_2 = 0$ nu ligger over 50%, ligesom sandsynligheden for $X_2 = 4$ ligger over 1 ‰):



1000 simuleringer



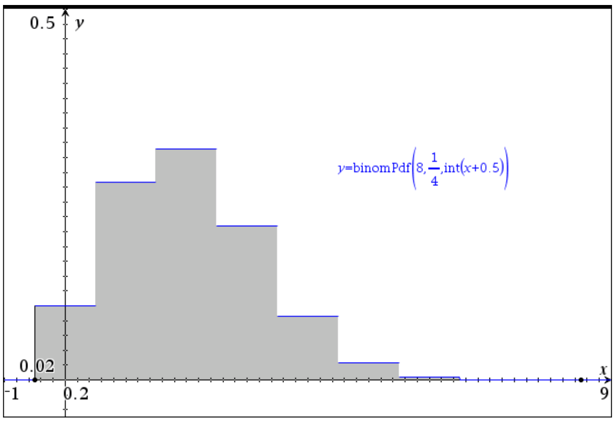
Teoretisk fordeling

Sammenhængen mellem binomialfordelingen og normalfordelingen

Nu hvor vi har mere styr over binomialfordelingerne kan vi begynde at studere formen på fordelingen lidt nærmere. Vi flytter da graferne over i **Graf**-værkstedet, idet vi opfatter punktsandsynligheden binomPdf(n,p,x) som en funktion af x. Det rejser dog det tekniske problem at x formelt skal være heltallig. Vi smører derfor de heltallige værdier ud over intervaller af længden 1, idet fx det hele tal 4 dækker intervallet fra 3.5 til 4.5. Teknisk set afbilder vi derfor grafen for funktionen

$$\mathrm{binom}(n, p, \mathrm{int}(x+0.5))$$

Her står int for heltalsdelen (integer part). Vi har netop lagt 0.5 til, så intervallet kommer til at ligge symmetrisk omkring det hele tal:



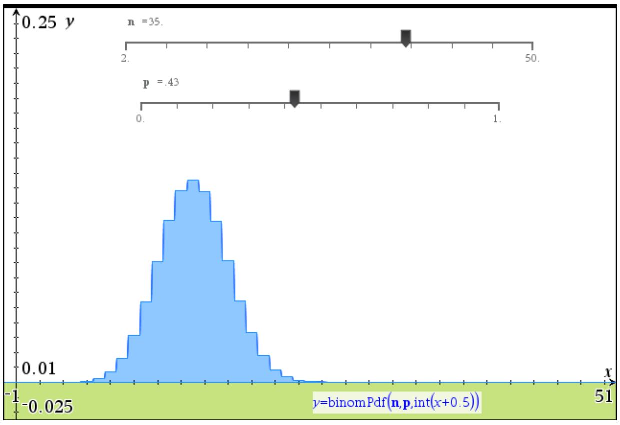
Vi har skraveret området under den stykvis konstante fordelingsfunktion ved at udregne integralet fra -0.5 til 8.5 (i **Undersøg Grafer**). Værdien af integralet er selvfølgelig 1, så det har vi efterfølgende skjult.

Grafen har den karakteristiske klokkeform, hvor sandsynligheden først vokser op til maksimumsværdien i $x=2$ og derefter falder ned til 0 igen.

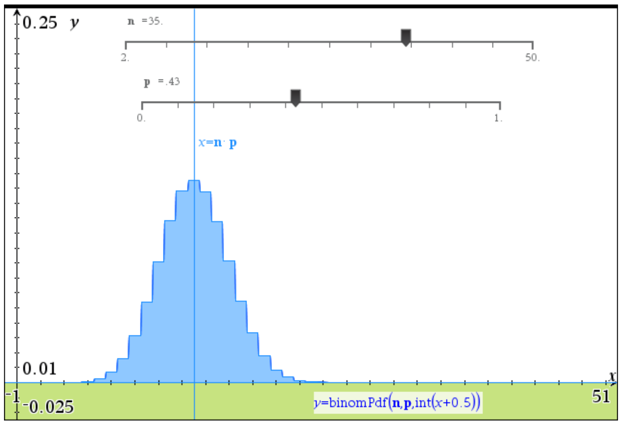
Hvis vi nu vil lege med sådanne fordelingsgrafer, kan det igen betale sig at oprette en dynamisk graf, hvor vi kan variere såvel antalsparameteren n (som vi sætter til at løbe fra 2 til 50) som sandsynlighedsparameteren p. Det gøres nemmest ved skydervariable.

Denne gang er det ikke så smart at bruge integralrutinen til at skravere med, da den vil sløve dynamikken alt for meget. I stedet tilføjer vi graferne for

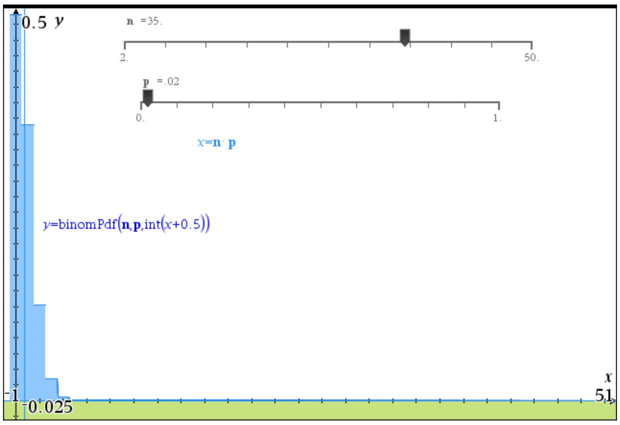
$$y \leq \text{binomPdf}(n, p, \text{int}(x+0.5)) \text{ og } y \leq 0$$



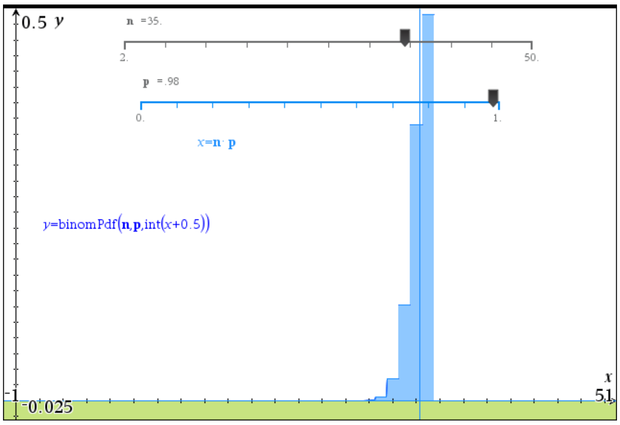
Dynamisk fordelingsgraf



Nu med middelværdi



højreskæv fordeling



venstreskæv fordeling

Trækker vi i skyderen for sandsynlighedsparameteren p kan vi netop se den karakteristiske klokkeformede graf på hele midterstykket. Men når vi trækker p for langt ud til siderne bryder klokken sammen og vi får i stedet for en monoton fordelingsgraf, sådan som vi også så det i forbindelse med Chevalier de Meres spil. Vi kan ydereligere tilføje middelværdien ved at indskrive $x = n \cdot p$ i en tekstboks og trække den ind på den ene af akserne. Trækker man i skyderen for p er det meget overbevisende at binomialfordelingen netop topper i middelværdien, som altså ikke blot angiver den forventede værdi, men også den mest sandsynlige værdi (vi præciserer dette udsagn i det følgende teoretiske afsnit).

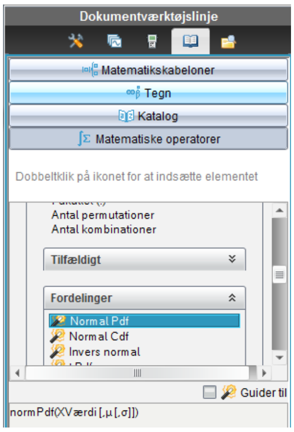
Men så længe vi holder os til midterstykket er der altså en tydelig klokkeformet fordeling. Det kunne godt minde om en normalfordeling, så det er fristende at lægge en normalfordeling ind over. Vi skal så skønne over normalfordelingens middelværdi μ og spredning σ. Men der er det jo nemmest at indtaste formlerne for binomialfordelingens middelværdi

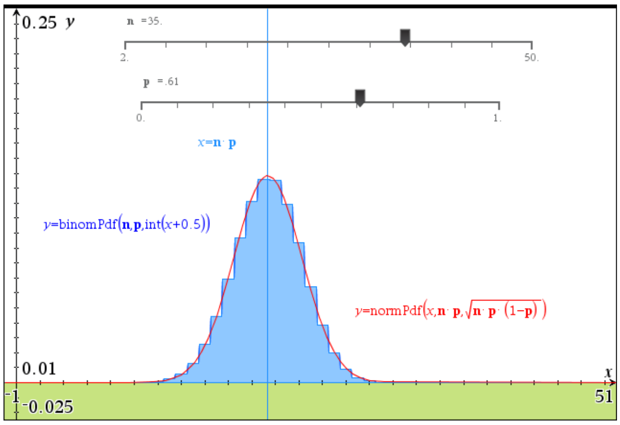
$$\mu = n \cdot p$$

og spredning

$$\sigma = \sqrt{n \cdot p \cdot (1-p)}$$

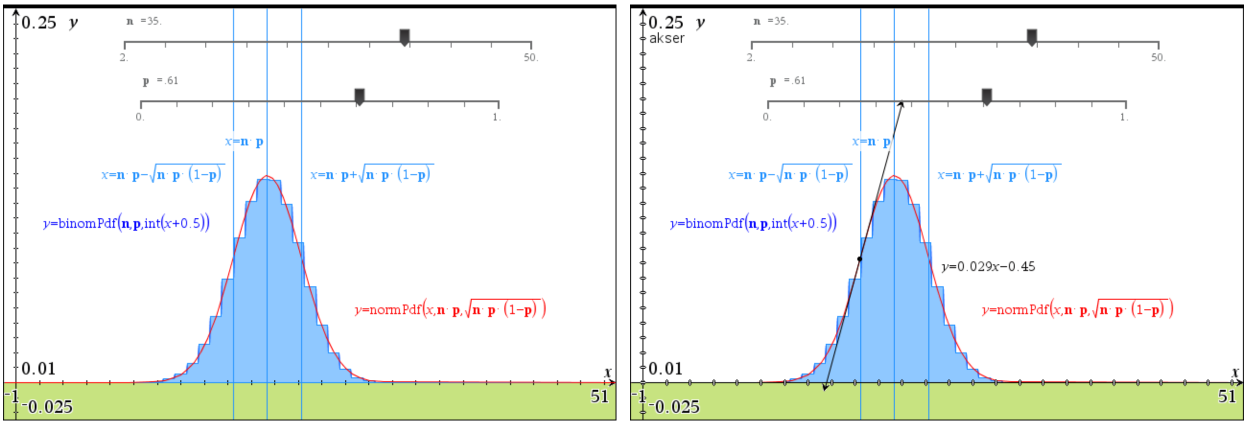
(ellers kan værdierne hentes fra en enkelt statistik udført på binomialfordelingen sådan som vi har set det i et tidligere afsnit):





Igen er den dynamiske visning meget overbevisende og man ser hvordan normalfordelingsgrafen smyger sig op af den klokkeformede binomialfordeling.

Vi kan endda også illustrere betydningen af spredningen σ. Hvor middelværdien svarer til centrum for fordelingen, så svarer spredningen til en form for radius for fordelingen. Ved at lægge én spredning til middelværdien eller trække én spredning fra middelværdien får vi derfor afgrænset det centrale område i klokken (med ca. 65% af observationerne)



Kigger vi nøje på grafen kan vi se at det centrale område netop afgrænser klokkens hat, dvs. det område hvor klokken er nedad hul.

Vi kan desværre *ikke* bekræfte dette ved at tilføje et vendepunkt fra **Undersøg graf**-menuen, for vi har jo ikke brugt de symbolske forskrifter for graferne. Men vi kan stadigvæk klistre en tangent på normalfordelingsgrafen og på den måde visuelt bekræfte at der synes at ligge en vendetangent i afstanden én spredning fra middelværdien.

Men alt i alt er det tydeligt at vi er ved at nå grænsen for hvad vi kan udrette med de *numeriske rutiner* til udregning af binomialfordlingen og normalfordelingen.

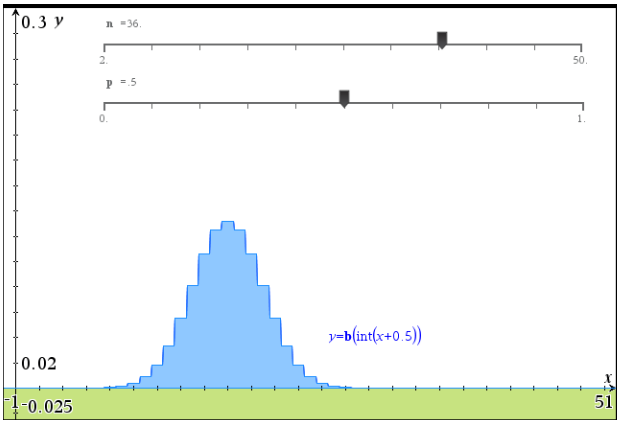
Binomialfordelingen som en eksakt fordeling: Monotoniforholdene*

Hvis man har en mere omfattende teori til rådighed for at arbejde med binomialfordelingen og normalfordelingen vil vi nu se på, hvad man kan bruge de eksakte forskrifter til. Specielt vil vi fokusere på, om vi rent faktisk kan stykke nogle beviser sammen for nogle af de observationer vi gjorde i det foregående. Vi vil da få stor hjælp af de symbolske udtryk.

For binomialfordelingen gælder der som bekendt:

$$\mathbf{b}(x):=\frac{n!}{x!\cdot(n-x)!}\cdot p^{x}\cdot(1-p)^{n-x} \text{ ▸ } \textit{Udført}$$

Vi har valgt at holde antalsparameteren n og sandsynlighedsparameteren p som ydre parametre, men de kunne selvfølgelig også være indbygget direkte i fordelingsfunktionen. For at kunne tegne grafen er vi derfor nødt til at tildele n og p værdier, hvilket sker nemmest som skydervariable lige som før. Vi har også det samme problem ved udregningen af funktionsværdier, at det er forudsat at x er en heltallig variabel. Men det klarer vi ligesom før ved at pakke argumentet ind i en int-funktion.



Så langt så godt, men det kunne vi jo også med den numeriske forskrift. Så hvor ligger fordelen ved den symbolske forskrift? Jo pointen er, at vi nu kan bevise at binomialfordelingen har de følgende karateristiske monotoniforhold

1. Binomialfordelingen vokser op fra værdien 0 til at maksium og aftager derefter igen til værdien 0.
2. Binomialfordelingen har maksimum tæt på middelværdien $\mu = n \cdot p$.

Hvordan kan vi nu vise dette? Da funktionen er stykvis konstant er det ikke så oplagt at bruge differentialregning. I stedet for skal vi tænke på den som en diskret funktion, der er defineret på hele tal. Vi er da intereseret i at se hvad der sker, når vi bevæger os fra $b(x)$ til $b(x+1)$. Traditionelt ville vi da kigge på *differensen* (svarende til hældningen). Men kig lige en gang på forskriften igen. Den består af en række faktorer, ikke af en række led. Og så er det faktisk mere nærliggende at se på fremskrivningsfaktoren. Vi flytter da først defintionen på binomialfordelingen over i en *ny* opgave, hvor vi kan få lov til at regne rent symbolsk (dvs. vi har ikke længere skydervariable til at tildele n og p numeriske værdier)!

$$\mathbf{b}(x):=\frac{n!}{x!\cdot(n-x)!}\cdot p^{x}\cdot(1-p)^{n-x} \text{ ▸ } \textit{Udført}$$

$$\mathbf{q}(x):=\frac{\mathbf{b}(x+1)}{\mathbf{b}(x)} \text{ ▸ } \textit{Udført}$$

$$\mathbf{q}(x) \text{ ▸ } \frac{p\cdot(x-n)}{(p-1)\cdot(x+1)}$$

Kan du se hvor de enkelte faktorer kommer fra?

Prøv først at overveje hvordan faktorerne p^x og $(1-p)^{n-x}$ bidrager til fremskrivningsfaktoren $q(x)$. Det kræver blot lidt kendskab til potensregneregler.

Prøv derefter at overveje hvordan faktorerne $x!$ og $(n-x)!$ bidrager til fremskrivningsfaktoren $q(x)$. Det kræver kendskab til reglen $(x+1)! = (x+1)\cdot x!$.

Overvej også at fremskrivningsfatoren $q(x)$ faktisk er positiv, selv om nogle af de faktorer, der indgår i udtrykket for $q(x)$ oplagt er negative ☺

Men som det ses kan CAS-værktøjet holde styr på reduktionen for os. Det er netop CAS-programmers styrke, at de kender reglerne for *den korrekte omskrivning af matematiske udtryk*. Til gengæld er det os, der skal lægge strategien for hvad udtrykkene skal bruges til. I modsætning til skakprogrammer, der ikke blot er i stand til at udføre trækkene korrekt, men også til i et vist omfang at tænke strategisk (ikke perfekt, men godt nok til at kun de allerbedste skakspillere i verden kan matche de bedste skakprogrammer), så kan CAS-programmer endnu ikke tænke strategisk. De mangler stort set kunstig intelligens. Så det er her vi selv kommer ind i billedet ☺.

Vi ved nu at monotoniforholdene for postive funktioner styres af fremskrivningsfaktoren:

Hvis $q(x) > 1$ så vokser funktionsværdien på stykket fra x til x+1.
Hvis $q(x) = 1$ så holdes funktionsværdien konstant på stykket fra x til x+1.
Hvis $q(x) < 1$ så aftager funktionsværdien på stykket fra x til x+1.

Første observation: Fremskrivningsfaktoren er en *aftagende* postiv funktion.

Det kan du faktisk se direkte af udtrykket, hvor vi udfører en blid omkrivning, så alle faktorerne er positive:

$$q(x) = \frac{(n-x)}{x+1} \cdot \frac{p}{1-p}$$

Det er kun den første brøk, der påvirkes af variationer i x. Og når argumentet x vokser, er det tydeligt at tælleren falder, ligesom nævneren vokser. Og det får netop brøken til at falde i værdi.

Men nu er brøkregningsargumenter i vore dage ofte sværere at forstå end differerential-regningsargumenter ☺. Så vi kan også bare se på differentialkvotienten for fremskriv-ningsfaktoren

$$\frac{d}{dx}(\mathbf{q}(x)) \;\blacktriangleright\; \frac{(n+1)\cdot p}{(p-1)\cdot (x+1)^2}$$

Her er tre af faktorerne oplagt positive, mens den sidste $(p-1)$ er oplagt negativ. Altså er differentialkvotienten negativ, og dermed er fremskrivningsfaktoren som påstået afta-gende. Det er netop denne observation, der rummer nøglen til forståelsen af binomial-fordelingens opførsel ☺

Lad os starte med at se på randværdierne for fremskrivningsfaktoren. Den første fremskrivningsfaktor er givet ved

$$\mathbf{q}(0) \;\blacktriangleright\; \frac{-n\cdot p}{p-1}$$

Den kan omskrives på formen

$$q(0) = \frac{n \cdot p}{1-p} = \frac{\mu}{1-p}$$

Hvis den første fremskrivningsfaktor ligger under 1 vil den derfor ligge under 1 hele vejen og binomialfordelingen vil derfor være stedse aftagende (højreskæv). Det kræver altså at middelværdien er så lille at den kommer under sandsynligheden for fiasko, dvs.

$$\mu < 1 - p$$

Den sidste fremskrivningsfaktor er givet ved

$\mathbf{q}(n-1) \blacktriangleright \frac{-p}{n\cdot(p-1)}$ ⚠

Den kan omskrives på formen

$$q(n-1) = \frac{p}{n-n\cdot p} = \frac{p}{n-\mu}$$

Hvis den sidste fremskrivningsfaktor ligger over 1 vil den derfor ligge over 1 hele vejen og binomialfordelingen vil derfor være stedse voksende (venstreskæv). Det kræver altså at middelværdien er så stor, at den kommer tættere på antalsparameteren n end sandsynligheden for succes, dvs.

$$\mu > n - p$$

I alle andre tilfælde ligger den første fremskrivningsfaktor over 1, den sidste under 1 og binomialfordelingen starter derfor med at vokse op til et maksimum, hvorefter den falder igen. Dette maksimum finder vi det sted hvor fremskrivningsfaktoren netop passerer 1:

$\mathrm{solve}(\mathbf{q}(x)=1,x) \blacktriangleright x=n\cdot p+p-1$ ⚠

Denne betingelse kan omskrives på formen

$$x_{\max} = n\cdot p-(1-p) = \mu-(1-p)$$

Hvis det tilfældigvis viser sig at være et helt tal er der to maksimumspunkter i binomialfordelingen, nemlig værdierne

$$x_1 = \mu-(1-p) \quad \text{og} \quad x_2 = \mu + p \qquad (\text{idet } q(x_1)=1, \text{ dvs. } b(x_1)=b(x_2))$$

Ellers vil der netop ligge ét helt tal i dette interval, hvor binomialfordelingen så topper. Det er da nemmest at bemærke, at den topper tæt på middelværdien, dvs. i praksis et af de to tal, der ligger på hver sin side af middelværdien, hvis ikke simpelthen middelværdien selv er et helt tal.

Konklusion:

a. En binomialfordeling er stedse aftagende, hvis den opfylder $\mu < 1-p$. I så fald topper binomialfordelingen i $x=0$.
b. En binomialfordeling er stedse voksende, hvis den opfylder $\mu > n-p$. I så fald topper binomialfordelingen i $x=n$.
c. I alle andre tilfælde er den først voksende, og dernæst aftagende, og den topper i det/de hele tal, der ligger i det lukkede interval $[\mu-(1-p);\mu+p]$, dvs. der er ét eller to maksimumspunkter afhængigt af om intervalgrænserne selv er hele tal. I praksis bemærker man, at hvis middelværdien selv er et helt tal, topper den altså i middelværdien. Ellers topper den i et af de to hele tal, der ligger på hver sin side af middelværdien.

Binomialfordelingen som en eksakt fordeling: Hulhedsforholdene*

Vi har nu styr på monotoniforholdene for binomialfordelingen. Kan vi gøre det samme med hulhedsforholdene og dermed retfærdiggøre klokkeformen? En sådan undersøgelse er nok ikke så velkendt, så det kan være klogt at kigge på et simpelt eksempel først:

$$\text{Normalfordelingen: } \phi(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2}$$

Den ved vi har en klokkeformet graf (the Bell curve) ☺. Vi udregner derfor første og anden afledede:

$$\varphi(x):=\frac{1}{\sqrt{2\cdot\pi}\cdot\sigma}\cdot e^{\frac{-1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2} \quad ▸ \textit{Udført}$$

$$\text{factor}\left(\frac{d}{dx}(\varphi(x))\right) ▸ \frac{-\sqrt{2}\cdot(x-\mu)\cdot e^{\frac{-x^2}{2\cdot\sigma^2}+\frac{\mu\cdot x}{\sigma^2}-\frac{\mu^2}{2\cdot\sigma^2}}}{2\cdot\sigma^3\cdot\sqrt{\pi}} ⚠$$

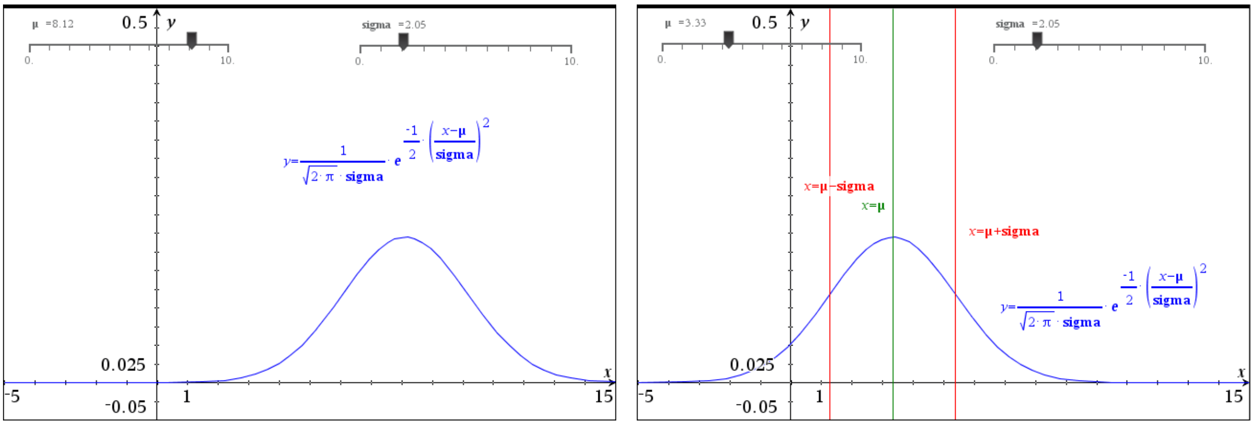
$$\text{factor}\left(\frac{d^2}{dx^2}(\varphi(x))\right) ▸ \frac{\sqrt{2}\cdot(x-\mu+\sigma)\cdot(x-\mu-\sigma)\cdot e^{\frac{-x^2}{2\cdot\sigma^2}+\frac{\mu\cdot x}{\sigma^2}-\frac{\mu^2}{2\cdot\sigma^2}}}{2\cdot\sigma^5\cdot\sqrt{\pi}} ⚠$$

Det kan godt virke lidt uoverskueligt, men eksponentialfunktionen er positiv, og de fleste faktorer er positive konstanter, så den interessante struktur ligger gemt i et førstegrads-polynomium for den første afledede og et andengradspolynomium for den anden afledede:

$$\phi'(x)=-(x-\mu)\cdot[\text{positivt udtryk}]$$
$$\phi''(x)=\big(x-(\mu-\sigma)\big)\cdot\big(x-(\mu+\sigma)\big)\cdot[\text{positivt udtryk}]$$

Overvej nu selv hvilke konskekvenser det har for monotoniforholdene henholdsvis hul-hedsforholdene for normalfordelingen.

Vi kan som sædvanlig illustrere det dynamisk ved at oprette en ny opgave, hvor vi i stedet håndterer middelværdien μ og spredningen σ som symbolske variable. Her løber vi så ind i det tekniske problem at σ er et lille græsk s, og da TI-Nspire CAS ikke skelner mellem store og små bogstaver svarer det til det store græske S, dvs. summationssymbolet Σ. Derfor må bogstavet σ ikke bruges som variabelnavn. Vi skriver det derfor bare ud som variablen sigma:

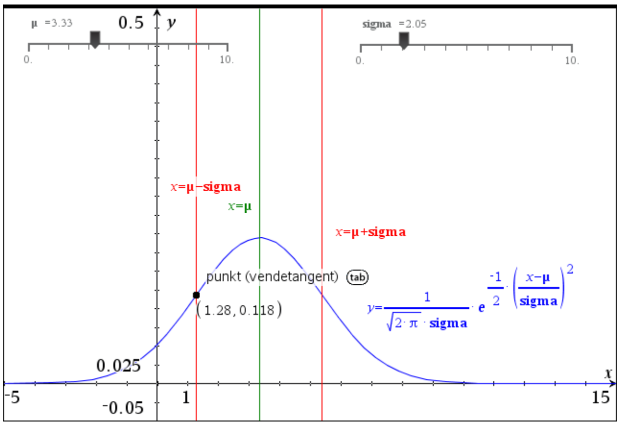


Vi kan da tydeligt se hvordan en variation af middelværdien μ blot fører til en vandret forskydning af grafen, mens en variation af spredningen sigma fører til en samtidig skalering af grafen (ud fra såvel symmetriaksen som *x*-aksen) – det samlede areal skal jo bevares ☺.

Det er også nemt som vist at tilføje markører for middelværdien μ og spredningen σ, ved blot at indskrive ligningerne $\boxed{x=\mu}$, henholdsvis $\boxed{x=\mu-\textbf{sigma}}$ og $\boxed{x=\mu+\textbf{sigma}}$ i tekstbokse, der efterfølgende trækkes ind på en akse.

Endelig kan vi bekræfte at én spredning netop svarer til skulderpunktet for klokken, altså vendepunktet, ved at finde vendepunktet ved hjælp af **Undersøg graf**-menuen:

6: Sandsynlighedregning

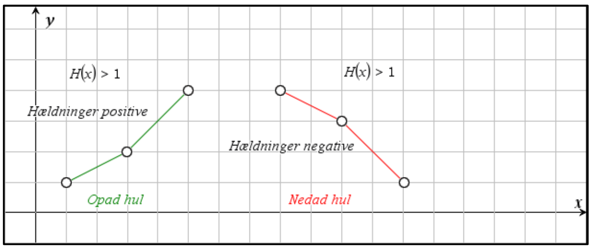


Vi vender os så mod binomialfordelingen. Denne gang er der tale om en stykvis konstant funktion, så vi kan ikke rigtigt komme til at anvende differentialregningen. I stedet må vi tænke i diskrete baner ☺

For at kontrollere hulheden må vi vide hvordan hældningen varierer. Hældningen for intervallet $[x;x+1]$ er givet ved $b(x+1)-b(x)$. Hældningen for det foregående interval er tilsvarende givet ved $b(x)-b(x-1)$. Ser vi på dobbeltintervallet rundt omkring *x*, dvs. $[x-1;x+1]$ er fremskrivningsfaktoren for hældningen fra det første delinterval til hældningen for det sidste delinterval derfor givet ved

$$H(x)=\frac{b(x+1)-b(x)}{b(x)-b(x-1)}$$

Det er denne hulhedsfunktion, der kontrollerer hulheden for os: Hvis fx den første hældning er positiv og $H(x)>1$ vokser hældningen, dvs. grafen er opad hul jfr. figuren nedenfor. Hvis til gengæld den første hældning er negativ og $H(x)>1$ aftager hældningen, dvs. grafen er nedad hul (jfr. figuren nedenfor) , osv.



Vi får altså brug for at vide hvornår $H(x)>1$, $H(x)=1$ og $0<H(x)<1$ og især det midterste tilfælde, dvs. $H(x)=1$ er interessant, fordi her er hældningerne ens, dvs. vi har fanget et vendepunkt!

Vi starter med at finde forskriften for fremskrivningsfaktoren for hældningen

$$\mathbf{b}(x):=\frac{n!}{x!\cdot(n-x)!}\cdot p^{x}\cdot(1-p)^{n-x} \blacktriangleright \textit{Udført}$$

$$\mathbf{h}(x):=\frac{\mathbf{b}(x+1)-\mathbf{b}(x)}{\mathbf{b}(x)-\mathbf{b}(x-1)} \blacktriangleright \textit{Udført}$$

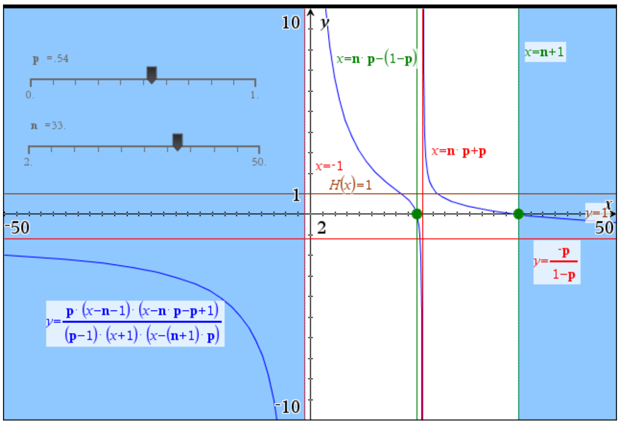
$$\mathbf{h}(x) \blacktriangleright \frac{p\cdot(x-n-1)\cdot(x-n\cdot p-p+1)}{(p-1)\cdot(x+1)\cdot(x-(n+1)\cdot p)}$$

Udtrykket er forbløffende simpelt (men ikke helt nemt at finde ved håndregning ☺). Igen består det af en række faktorer, der er oplagt positive/negative, men også et par faktorer, der kan være lidt sværere at gennemskue. Vi omskriver en smule på forskriften og finder

$$H(x)=\frac{p}{1-p}\cdot\frac{n+1-x}{x+1}\cdot\frac{x-(\mu-(1-p))}{x-(\mu+p)}$$

De første fire faktorer er trivielt positive i intervallet $[0;n]$. Men de to sidste faktorer skifter faktisk fortegn undervejs. De følges dog næsten ad, idet tælleren skifter fortegn i $x_1=\mu-(1-p)$, mens nævneren skifter fortegn i $x_2=\mu+p$. Men det svarer jo netop til grænserne for det interval af længde 1, hvor vi finder maksimumspunktet. Bortset fra dette interval er fremskrivningsfaktoren for hældningen derfor positiv!

En graf bekræfter dette, hvor vi også har tilføjet nulpunkterne $x=\mu-(1-p)$, $x=n+1$, de lodrette asymptoter $x=-1$, $x=\mu+p$, den vandrette asymptote $y=-\dfrac{p}{1-p}$ samt den kritiske linje $H(x)=1$.



Det er området mellem de to lodrette asymptoter, der bestemmer hulhedsforholdene for binomialfordelingen. Til venstre for maksimumsintervallet

$$[\mu-(1-p);\mu+p]$$

ved vi hældningerne er positive. Da fremskrivningsfaktoren for hældningen starter med at ligge over 1 er grafen først opad hul. Men så krydser vi den kritiske linje $H(x)=1$, hvor der er et vendepunkt, og på resten af stykket hen til maksimumsintervallet er grafen nedad hul. På den anden side af maksimumsintervallet sker det samme! Vi kan altså begrunde klokkeformen i stor detalje.

Vi ser yderligere at der må være netop to vendepunkter, et på hver sin side af maksimums-intervallet. Vi kan endda nemt finde formlerne for vendepunkterne:

$$\text{solve}(h(x)=1,x) \blacktriangleright x=\frac{-\left(\sqrt{-4\cdot n\cdot p\cdot(p-1)-4\cdot p^2+4\cdot p+1}-2\cdot n\cdot p-2\cdot p+1\right)}{2} \text{ or } x=\frac{\sqrt{-4\cdot n\cdot p\cdot(p-1)-4\cdot p^2+4\cdot p+1}+2\cdot n\cdot p+2\cdot p-1}{2}$$

Det kan se lidt uoverskueligt ud, men flytter vi nævneren 2 op i tælleren letter tågen:

$$v_1=n\cdot p+p-\tfrac{1}{2}-\sqrt{n\cdot p\cdot(1-p)-p^2+p+\tfrac{1}{4}}=\mu+p-\tfrac{1}{2}-\sqrt{\sigma^2-p^2+p+\tfrac{1}{4}}$$
$$v_2=n\cdot p+p-\tfrac{1}{2}+\sqrt{n\cdot p\cdot(1-p)-p^2+p+\tfrac{1}{4}}=\mu+p-\tfrac{1}{2}+\sqrt{\sigma^2-p^2+p+\tfrac{1}{4}}$$

Det første led er netop midtpunktet af maksimumsintervallet, dvs. symmetripunktet for binomialfordelingen og det ligger tæt på middelværdien μ. Det andet ligger tæt på spredningen σ. Den diskrete binomialfordeling opfører sig altså i det store og hele næsten som normalfordelingen ☺

7 Hypotesetest

I dette kapitel vil vi se på nogle grundlæggende eksempler på, hvordan man udfører en simpel eksperimentel hypotesetest, først en **goodness-of-fit-test** og dernæst en **uafhængighedstest**. Herunder vil vi stifte bekendtskab med de grundlæggende generelle begreber. Derefter vil vi knytte eksemplerne til den specielle χ^2-test (læses: chi2-test) og dermed bruge de samme eksempler som en indføring i χ^2-testen.

Terningekast

Vi antager at vi forud for deltagelsen i et terningespil får lov til at kaste 24 gange med terningen for at registrere antallet af seksere. Vi *observerer* nu, at vores prøverunde giver os 10 seksere. Skal denne *observation* nu give anledning til bekymring? Tyder den på, at terningen er skæv og derfor viser for mange seksere?

Umiddelbart *forventer* vi at terningen giver lige mange ettere, toere, treere, firere, femmere og seksere. Denne tro på at udfaldet af terningekastene er rent tilfældigt og ikke indeholder en systematisk skævvridning af de enkelte udfald kaldes *nulhypotesen* H_0. På basis af nulhypotesen *forventer* vi, at vi ud af 24 kast finder fire seksere.

På den anden side er terningekast et stokastisk fænomen, så i praksis kan vi ikke regne med netop fire seksere, somme tider vil vi få flere, sommetider færre, men i *middel* forventer vi fire seksere, hvis vi gentager de 24 terningekast rigtigt mange gange med en *fair terning*, der opfører sig som forventet ud fra nulhypotesen.

Uden en *nulhypotese* H_0, der generelt udsiger at udfaldene skyldes rene tilfældigheder, kan vi *ikke* udtale os om hvad man må *forvente* af udfaldene i et stokastisk fænomen.

Nu observerede vi netop 10 seksere og det ligger et pænt stykke over de forventede fire seksere: Hvad kan vi konkludere ud fra det? Er forskellen så stor, at det ikke længere er rimeligt at tro på observationen som et resultat af de uundgåelige tilfældige udsving i den enkelte prøverunde? Eller er det så nemt at få en så stor forskel, når man kaster med en fair terning, at det ikke er rimeligt at forkaste vores tro på nulhypotesen om en fair terning?

For at kunne besvare dette spørgsmål på rimelig vis må vi først undersøge hvor nemt det er at få en afvigelse, der er mindst lige så stor, under forudsætning af at nulhypotesen holder; dvs. hvor nemt er det at få mindst 10 seksere ved kast med en fair terning.

Dette spørgsmål kan undersøges rent teoretisk, men det kan også undersøges rent eksperimentelt ved *en simulering af nulhypotesen*. Det er den sidste vej vi her vil følge.

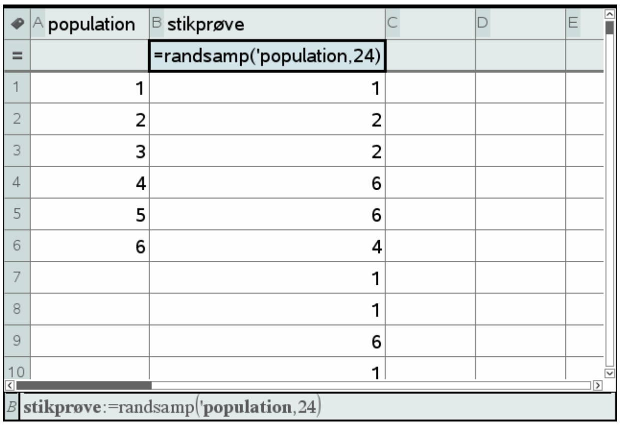
Simulering af nulhypotesen

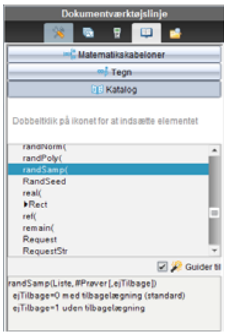
Man kan simulere nulhypotesen på mange måder, men vi vil her følge den samme strategi som i kapitlet om sandsynlighedsregning og tage udgangspunkt i en **population**, der følger den ideelle fordeling i overensstemmelse med nulhypotesen.

Da alle udfaldene er *lige sandsynlige* skal de altså forekomme lige mange gange i populationen. Det nemmeste er da at oprette en population, der netop rummer en ”etter”, en ”toer”, en ”treer”, en ”firer”, en ”femmer” og en ”sekser”. Som vi denne gang indskriver som tal.

$$\textbf{population} := \{1,2,3,4,5,6\}$$

Vi indskriver derfor populationen som vist i et **Lister og regneark**-værksted.



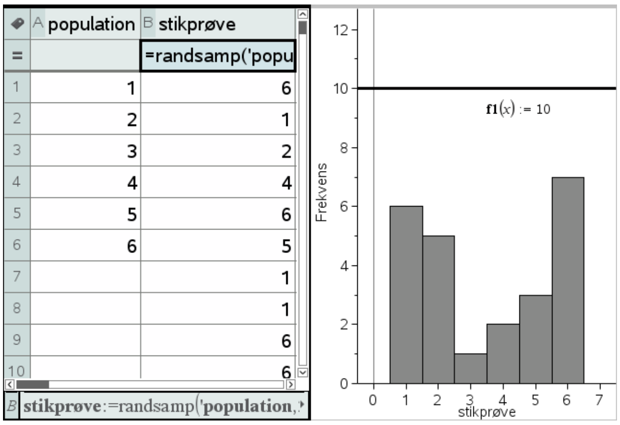


Vi indskriver tilsvarende en stikprøve, der netop rummer 24 tilfældigt udvalgte elementer fra populationen, idet stikprøve udtages *med tilbagelægning*, så hver eneste udtrækning har *samme sandsynlighed* for at trække en etter, en toer, … og en sekser. Det sker ved hjælp af kommandoen

= randSamp(**population**,24),

der som standard netop foregår *med* tilbagelægning. Stikprøven svarer altså til de 24 terningekast *under forudsætning af at nulhypotesen holder*!

Vi kan gentage simuleringen ved at taste **CTRL/CMD R** inde i regnearket. På den måde kan vi nu eksperimentelt undersøge, hvor nemt det er at få ti seksere ved 24 kast med en fair terning.



Vi skal da have fremstillet udfaldene grafisk, så vi kan danne os et overblik over strukturen af de 24 terningekast. Vi opretter derfor et **Diagrammer og statistik**-vindue og afsætter stikprøven som et histogram. Yderligere tilføjer vi grafen for den konstante funktion 10, der fungerer som en overligger. Hvis diagrammet kommer op og røre overliggeren, eller lige frem bryder igennem overliggeren, så er det lykkedes os frembringe et udfald, der er *mindst lige så skævt som det observerede.*

Proto-test og signifikansniveau

Vi er nu næsten klar til at udføre en første primitiv test, en *proto-test*, for at teste nulhypotesens troværdighed. Men før vi udfører proto-testen må vi imidlertid træffe et afgørende valg: Hvor svært skal det være at frembringe et skævt udfald, før vi forkaster nulhypotesen og i stedet foretrækker den ret upræcise alternative hypotese om at der udover tilfældighederne også foreligger en eller anden form for systematisk variation, der skævvrider udfaldene. Det niveau, der fører til forkastelse af nulhypotesen kaldes *signifikansniveauet*. Det bygger på en ren konvention, der har vist sig nyttig i praksis. Ligegyldig hvor mange seksere vi observerer i prøverunden kan vi nemlig aldrig med sikkerhed afvise nulhypotesen. Men nulhypotesen kan blive en så utroværdig forklaring på det observerede, at det kan betale sig at tro på den alternative hypotese, velvidende at vi løber en lille, men ikke forsvindende risiko, for at tage fejl.

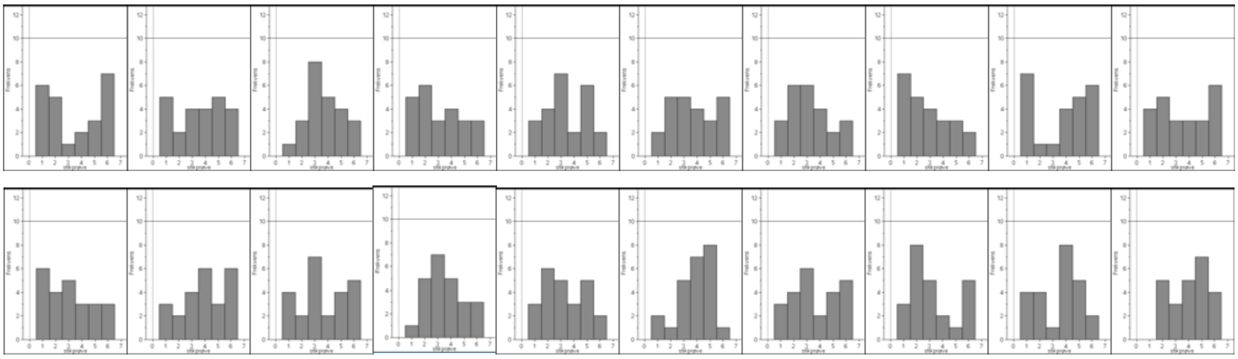
I praksis har man typisk valgt signifikansniveauet 5%, men under specielle omstændigheder kan denne procentdel sættes op eller sættes ned efter behov. Her vil vi dog holde fast i de 5%.

Sandsynligheden for at få et *skævt udfald* (mindst 10 seksere), når man kaster 24 gange med en fair terning (nulhypotesen) kaldes *p*-værdien.

Hvis p-værdien falder under signifikansniveauet på 5%, dvs. hvis de skæve udfald forekommer i færre end hver tyvende prøverunde, regnes nulhypotesen for så utroværdig, at den forkastes.

For at danne os et skøn over p-værdien kan vi derfor starte med at udføre simuleringen 20 gange. Hvis der ikke forekommer et eneste skævt udfald, tyder det på at p-værdien ligger under 5%, hvorfor nulhypotesen må forkastes. Hvis der omvendt forekommer et eller flere skæve udfald, tyder det på at p-værdien ligger over 5%, hvorfor vi ikke kan forkaste nulhypotesen, der derfor ind til videre står til troende ☺

Vi gentager derfor simuleringen 20 gange og tjekker om vi rammer overliggeren for sekserne.



Som det ses, lykkedes det ikke en eneste gang for sekserne at ramme overliggeren. Der var altså ingen skæve udfald i vores proto-test. Skønnet for p-værdien er derfor, at p-værdien ligger *under* 5%, hvorfor det ser ud til at vi må forkaste nulhypotesen.

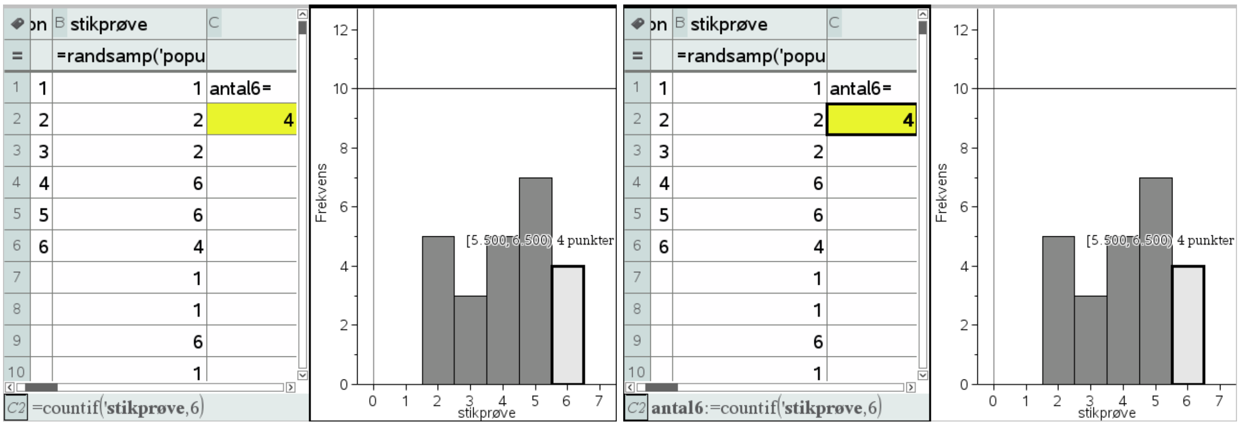
Men 20 simuleringer er selvfølgelig et spinkelt grundlag at drage vidtrækkende konklusioner på ☺. Så vi ser nu på, hvordan vi kan sætte antallet af simuleringer i vejret på en systematisk måde.

Testværdi og datafangst med LUA-kommandoen *capture*

Vi har hidtil forladt os på en visuel afgørelse af om et udfald er skævt. Vi vil nu også *regne* på sagen: Vi skal da først indføre en formel, der tæller antallet af seksere, som er den *testværdi*, der skal bruges til at afgøre om en simulering giver anledning til et *skævt udfald*, dvs. et udfald, hvor antallet af seksere er mindst 10. Det sker nemmest i en celle med brug af **countif**-kommandoen: I celle C1 skriver vi derfor teksten ”antal6 =” og i celle C2 formlen

$$\boxed{C2} = \text{countif}(\text{'}\mathbf{stikprøve}, 6)$$

Som det ses er der overensstemmelse mellem beregningen og den grafiske optælling.



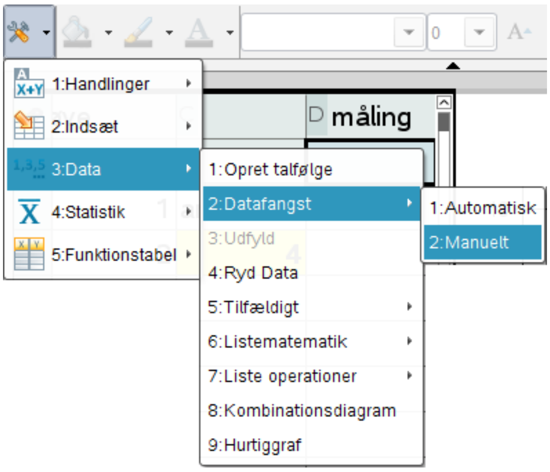
Vi kan nu opsamle antallet af seksere ved at gemme dem som en variabel, dvs. vi omskriver celleformlen til

$$\boxed{C2}\ \mathbf{antal6} := \text{countif}(\text{'}\mathbf{stikprøve}, 6)$$

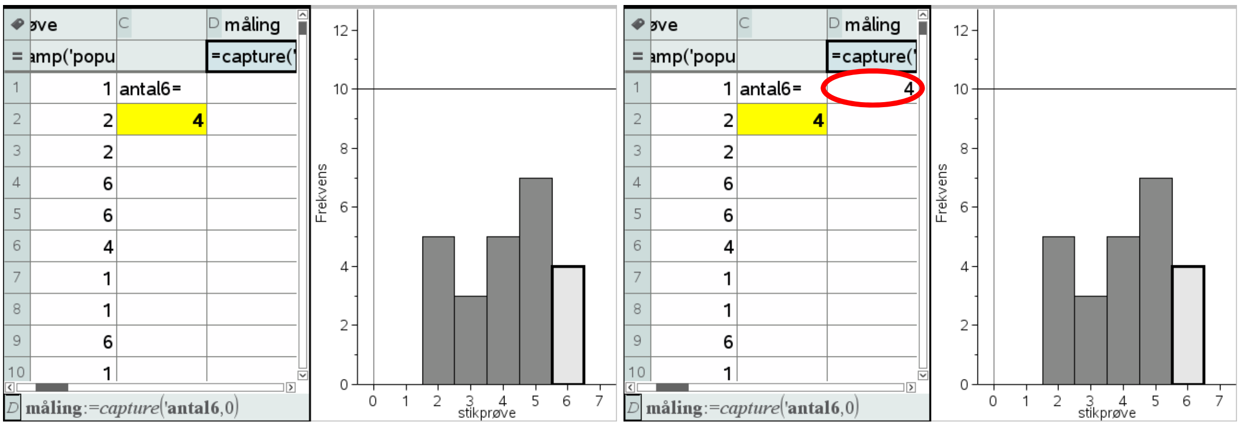
Antal seksere er nu lagret som en variabel. Læg specielt mærke til, at tallet nu skrives i fed for at markere, at det er lagret som en variabel.

Vi kan derfor udføre en datafangst ved hjælp af LUA-kommandoen *capture*. Det er samme teknik, som vi allerede har beskrevet i kapitel 6 om sandsynlighedsregning. Vi opretter derfor en ny søjle kaldet **måling**, og vælger den følgende kommando i **Data**-menuen

Obs
En **datafangst** med capture-kommandoen overvåger en variabel i variabelregistret. En **automatisk** datafangst registrerer hver gang variabel skifter værdi. En **manuel** datafangst registrerer variablens værdi når datafangsten udløses med **CTRL/CMD punktum**.



Her vælger vi en **Manuel datafangst**, da antallet af seksere jo ikke nødvendigvis skifter værdi, bare fordi vi udfører en ny simulering. Resultatet er den følgende formel



Obs
Capture-kommandoen virker kun i **Lister og regneark**-værkstedet. Det er jo netop *ikke* en matematik-kommando, men en LUA-kommando. Derimod virker udløsningskommandoen **CTRL/CMD .** fra ethvert værksted.

Læg mærke til, at *capture*-kommandoen står i kursiv – det er netop *ikke* en matematik-kommando, men en LUA-kommando. Inde i denne *capture*-kommando skal du nu angive hvilken variabel vi skal fange, her variablen **antal6**. Endelig er der en parameter 0, der fortæller LUA-proceduren, at der er tale om en manuel datafangst.

Læg også mærke til, at der i første omgang slet ikke sker noget! Men det er jo fordi der er tale om en manuel datafangst, dvs. vi skal selv udløse datafangsten. Det sker ved at taste **CTRL/CDM .** (Kontrol Punktum). Det er ligegyldigt, hvor vi befinder os, når vi taster **CTRL/CMD .** – den manuelle datafangst udløses altid af denne taste-kombination, også selvom vi befinder os i et helt andet værksted.

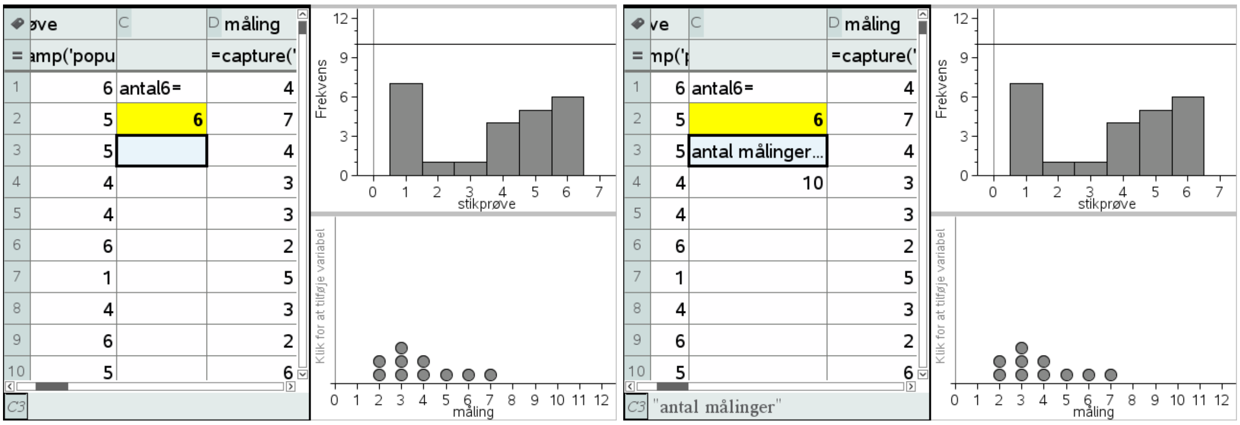
Nu skete der så endeligt noget! Vi har fanget antallet af seksere i den første simulering ☺

Obs
Husk at stå i **Lister og regneark**-værkstedet, for det er kun her at **CTRL/CMD R** udløser en genberegning af regnearket og dermed en ny simulering.

Så skal vi holde tungen lige i munden. Vi skal nu stille os i **Lister og regneark-**værkstedet og skiftetvis taste **CTRL/CMD R** for at udføre en ny simulering henholdsvis **CTRL/CMD .** for at udløse en ny datafangst!

Det kan godt lyde lidt indviklet men med lidt øvelse kan man fx holde **CTRL/CMD-**tasten ned med venstre hånd, mens man med højre hånd skiftevis taster **R** henholdsvis **Punktum**.

Resultatet af de første 10 simuleringer kan fx se således ud:



Inden vi går i gang med at lave datafangst i større omfang vil vi lige klargøre datafangsten, så vi dels kan se hvor mange målinger vi har foretaget, dels kan følge opbygningen af fordelingen for disse målinger.

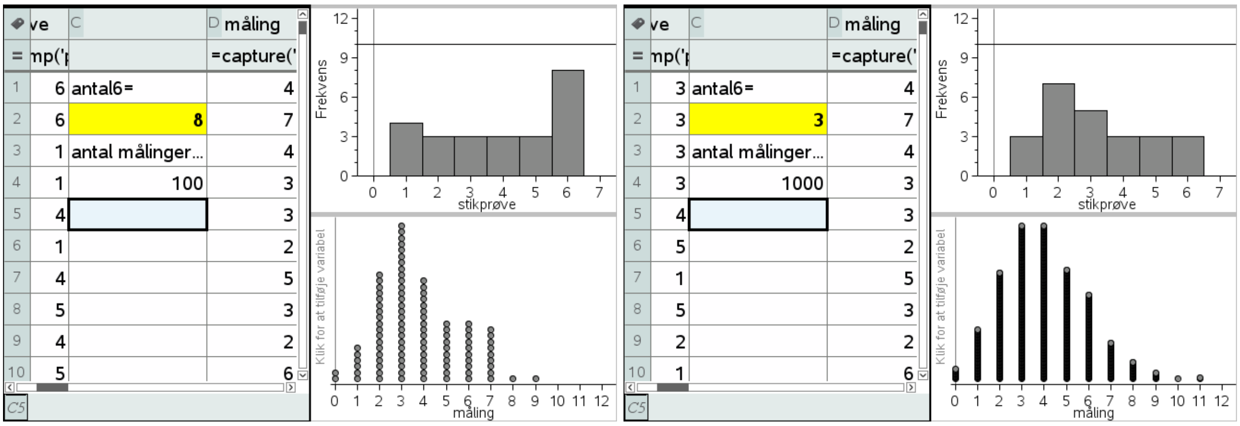
Antallet af målinger fanger vi med celleformlen

$$\boxed{C4} = \text{count}('\textbf{måling})$$

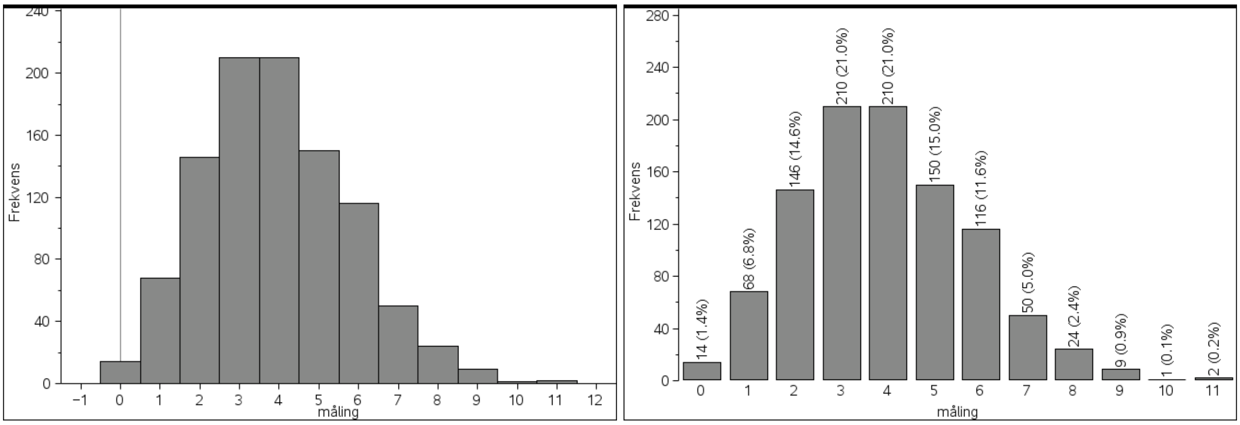
Visualiseringen af fordelingen klarer vi ved at oprette et nyt **Diagram og statistik-**vindue diagramtype til et prikdiagram for variablen **måling**.

Vi er så klar til at danse med fingrene, idet vi først taster **CTRL/CMD R** og derefter taster **CTRL/CMD punktum**.
Vi viser først resultatet af de første 100 målinger, derefter resultatet af de første 1000 målinger.

Der tegner sig da et mønster, hvor det hyppigste udfald som forventet er fire seksere, om end det holdt hårdt – på lange strækninger var det 3, der var det hyppigste udfald, og efter 1000 udfald i denne tilfældige serie, endte de med at være lige hyppige. Men der er også en hel del udfald med fra 0 til 9 seksere og faktisk også nogle få skæve udfald med 10 og 11 seksere.



Vi kan danne os et bedre indtryk af fordelingen ved at skifte diagramtype til et histogram eller et søjlediagram (hvor vi så må tvinge målingen til at være en kategorisk variabel). I det sidste tilfælde kan vi så også slå **Vis alle etiketter** til:



I vores tilfælde ser vi da, at et skøn over p-værdien er givet ved

$$p \approx \frac{3}{1000} = 0.3\%$$

Man kan selvfølgelig med rette synes det er bøvlet at gennemføre den ovenstående simulering 1000 gange.

Det ville da også være rart hvis man kunne flytte den over på en automatisk datafangst, der er meget nemmere at udføre i praksis. Vi har da det problem, at hvis målingen gentager sin værdi, registreres den *ikke* automatisk. Der findes snedige tricks til at omgå dette problem, men de er netop snedige og vi vil derfor ikke gennemgå dem i denne eksempelsamlingen. Men der findes fx et hæfte om **Statistik med TI-Nspire CAS**, hvor man kan finde detaljerne.

Avanceret datafangst med sekvens-kommandoer*

Vi vil nu se på mulighederne for at udføre datafangsten i ét hug, dvs. udregne antallet af seksere for 1000 simuleringer og opbygge en liste med de fundne værdier i ét hug, i stedet for trinvis, som vi så det med LUA-kommandoen *capture*. Hertil bruges sekvenskommandoen, der frembringer en talfølge ud fra et udtryk

$$\text{seq}(\mathit{udtryk}, \mathit{indeks}, \text{startværdi}, \text{slutværdi})$$

Man kan sagtens arbejde med sekvens-kommandoen seq i **Lister og regneark**, men ved mere komplicerede sekvenskommandoer er det mere overskueligt at flytte arbejdet over i et **Note**-værksted. Her viser vi først hvordan kommandoen virker ved at udregne en liste over de 10 første kvadrattal:

$$\text{seq}(x^2, x, 1, 10) \blacktriangleright \{1,4,9,16,25,36,49,64,81,100\}$$

Men ellers går vi frem præcis ligesom før. Først oprettede vi en **population**, dernæst trak vi en **stikprøve** på fireogtyve elementer (med tilbagelægning!) ved hjælp af RandSamp-kommandoen. Denne kommando er helt central i sekvenskommandoen, da det er den, der styrer dynamikken og skal sikre, at vi får et nyt antal seksere i hver simulering. Endelig benyttede vi en Countif-kommando til at finde antallet af seksere i stikprøven. I første omgang tester vi nu vores sekvens-kommando ved blot at opsamle 10 værdier:

Først henter vi lige populationen

population ▸ {1,2,3,4,5,6}

Så opstiller vi den formel, der udregner stikprøven

randSamp(**population**,24) ▸ {5,4,4,5,3,6,1,5,3,2,2,2,5,3,4,1,4,3,4,6,4,3,2,3}

Så tæller vi hvor mange seksere, der var ved at bruge kommandoen

countif(*liste*,6)

hvor *liste* erstattes af formlen for stikprøven

countIf(randSamp(**population**,24),6) ▸ 8

Endelig frembringer vi talfølgen ved hjælp af sekvenskommandoen

seq(*udtryk*,x,1,10)

hvor *udtryk* erstattes af den ovenstående tælle-kommando

seq(countIf(randSamp(**population**,24),6),*x*,1,10) ▸ {1,6,5,6,4,4,2,2,2,4}

Som det kan ses virker seq-kommandoen! Hvis man indskriver den direkte i ét hug inde i fx et **Lister og regneark**-værksted, skal man være meget forsigtigt med parenteserne, da det udtryk, der frembringer talfølgen, er sammensat af flere lag:

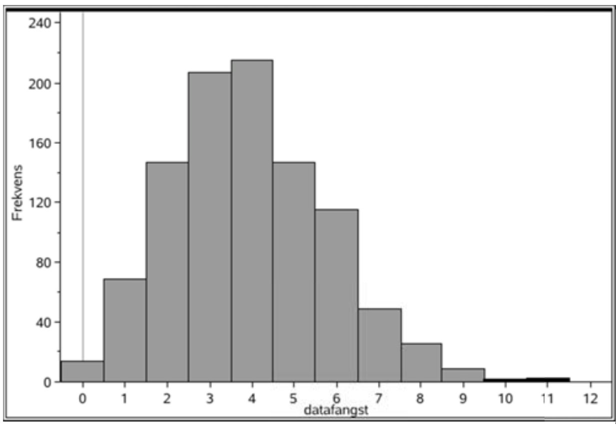
$$\text{seq}\left(\boxed{\text{countif}\left(\boxed{\text{randSamp}(\textbf{population},24)},6\right)},x,1,10\right)$$

Det yderste lag består af tælle-kommandoen, mens det inderste lag består af stikprøvekommandoen.

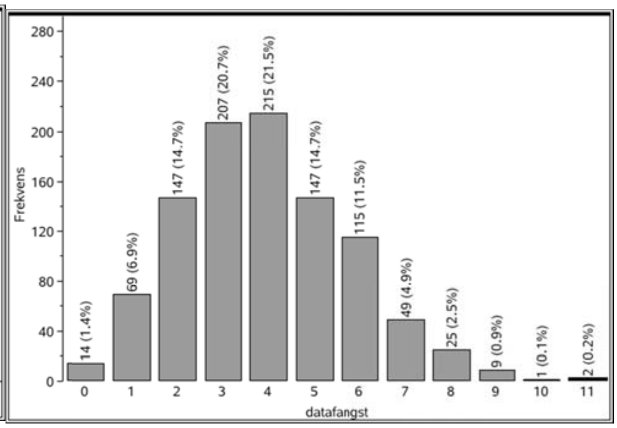
Men når først det virker kan vi jo roligt navngive listen og sætte antallet af gentagelser op til 1000:

datafangst:=seq(countIf(randSamp(**population**,24),6),*x*,1,1000)
▸ {4,4,4,3,3,1,2,3,3,4,4,3,2,5,3,4,1,7,5,9,2,5,2,6,7,4,9,4,2,4,2,3,4,5,5,2,1,0,4,5,5,6,4,4,5,1,6,6,4

Der går et splitsekund og vi har hentet de 1000 målinger! For at kunne overskue dem kan vi derefter afbilde dem i et **Diagrammer og Statistik**-værksted



histogram (numerisk variabel)



søjlediagram (kategorisk variabel)

I begge tilfælde ser vi da igen, at sandsynligheden for at få et skævt udfald er langt under 5%. Denne gang er vores skøn over p-værdien baseret på 1000 gentagne simuleringer igen givet ved 0.3%.

Sandsynligheden for at få mindst 10 seksere i fireogtyve forsøg synes altså stadig at ligge langt under signifikansniveauet, hvorfor nulhypotesen forkastes som utroværdig!

Man kan synes det er lidt kompliceret med de indpakkede sekvenskommandoer. Hvis man kender lidt til programmering er det da også muligt at oprette en brugerdefineret funktion, med en langt mere overskuelig opbygning. Men i dette hæfte vil vi ikke komme nærmere ind på programmering med TI-Nspire CAS. Det kan man i stedet læse mere om i et særskilt hæfte om **Programmering med TI-Nspire CAS**.

Den teoretiske p-værdi (Binomialfordelingen)*

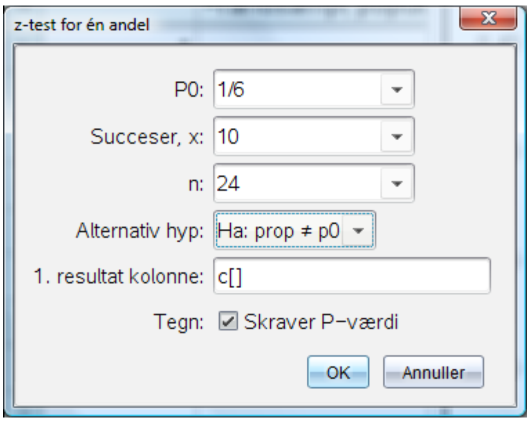
Hvis du kender lidt til binomialfordelingen er det nu ikke så svært at udregne den eksakte p-værdi. Antallet af seksere X er jo ifølge nulhypotesen binomialfordelt med antalsparameter $n = 24$ og sandsynlighedsparameter $p = 1/6$. Vi kan derfor bruge intervalsandsynlighedsfunktionen binomCdf til at udregne den ønskede sandsynlighed $p(X \geq 10)$:

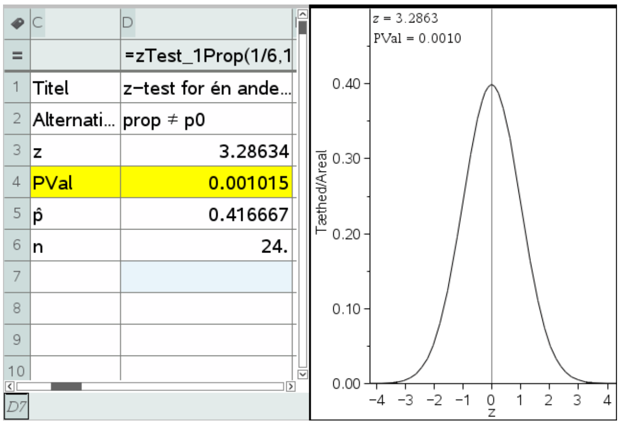
$$\text{binomCdf}\left(24, \frac{1}{6}, 10, 24\right) \blacktriangleright 0.003339$$

Den eksakte p-værdi er altså givet ved 0.3339% i god overensstemmelse med de skøn vi fandt ved at simulere.

Vi kan endda få TI-Nspire CAS til at udføre det indbyggede kanoniske test for at teste nulhypotesen. Det hedder en **z-test for én andel** (idet der benyttes normalfordelingsapproksimation til binomialfordelingen for at udregne p-værdien, dvs. testet udføres reelt som et normalfordelingstest). Andelen står for antal succeser i forhold til det samlede antal forsøg. Antal succeser er netop den observerede værdi.

Vi tester nulhypotesen at terningen er fair, dvs. at sandsynligheden for succes netop er 1/6. Den alternative hypotese er da at sandsynligheden for succes er forskellig fra 1/6. I binomialtestet kan man dog også teste ensidigt, dvs. nulhypotesen kunne fx være at sandsynligheden for succes er højst 1/6, hvorved den alternative hypotese vil blive at sandsynligheden for succes er større end 1/6.





Ved den kanoniske test finder vi altså det følgende skøn over p-værdien 0.1%. Det er noget lavere end den eksakte værdi på 0.33%. Fejlen skyldes først og fremmest *diskretiseringsfejlen* (som er indbygget i alle de kanoniske tests).

Normalfordelingsapproksimationen bygger sit skøn på intervallet $[10;\infty[$. Her gør det ikke så meget at vi udstrækker halen til uendelig, men binomialfordelingen er diskret og udfaldet 10 dækker derfor over intervallet [9.5;10.5[, hvorfor normalfordelings-approksimationen burde bygge sit skøn på intervallet $[9.5;\infty[$.

Normalfordelingsapproksimationen bygger endvidere på normalfordelingen med samme middelværdi og spredning som binomialfordelingen:

$$\mathbf{n}:=24 \blacktriangleright 24 \qquad \mathbf{p}:=\frac{1}{6} \blacktriangleright \frac{1}{6}$$

$$\mathbf{\mu}:=\mathbf{n}\cdot\mathbf{p} \blacktriangleright 4 \qquad \mathbf{sigma}:=\sqrt{\mathbf{n}\cdot\mathbf{p}\cdot(1-\mathbf{p})} \blacktriangleright \frac{\sqrt{30}}{3}$$

Den kanoniske p−værdi:

$$2\cdot \mathrm{normCdf}(10,\infty,\mathbf{\mu},\mathbf{sigma}) \blacktriangleright 0.001015$$

Den forbedrede p−værdi

$$2\cdot \mathrm{normCdf}(9.5,\infty,\mathbf{\mu},\mathbf{sigma}) \blacktriangleright 0.002591$$

Det er jo noget bedre, idet skønnet nu kom op på 0.25%, om end det stadigvæk ikke er perfekt.

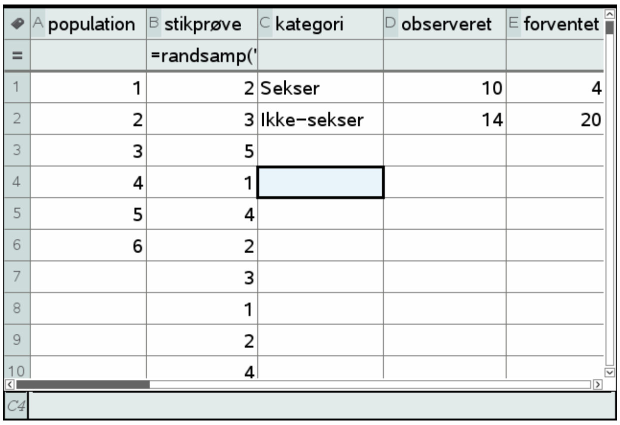
Man kan selvfølgelig undre sig over, at man overhovedet bruger normalfordelings-approksimationen, når man lige så godt kunne regne på den eksakte binomialfordeling. Det skyldes tradition! De kanoniske tests blev udviklet *før* computeren, og den gang var det ret kompliceret at håndtere vilkårlige binomialfordelinger, med de mange valgmuligheder for parametrene n og p, mens normalfordelingen kun krævede adgang til en god tabel over standardnormalfordelingen med middelværdi 0 og spredning 1. Altså blev normalfordelingsapproksimationen enerådende, og i dag har man vænnet sig til at bruge den også selv om det strengt taget ikke er nødvendigt. Hypotesetest er under alle omstændigheder *ikke* nogen eksakt videnskab, men bygger på en række mere eller mindre vikårlige konventioner, fx valget af signifikansniveauet. Derfor er det ikke så afgørende at få fat i den eksakte p-værdi. Bare alle parter er enige om beregnings-metoden kan man jo få truffet det afgørende valg, *om man skal forkaste nulhypotesen eller ej*, på et rimeligt og brugbart grundlag ☺

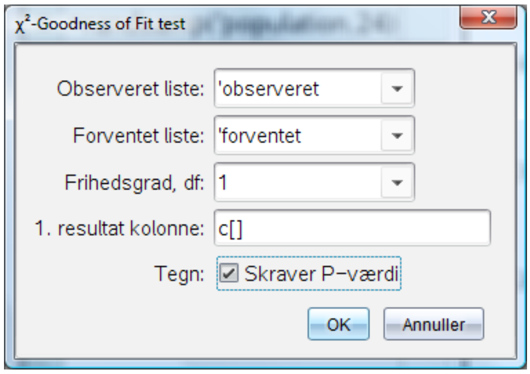
Terningekast som χ^2-test

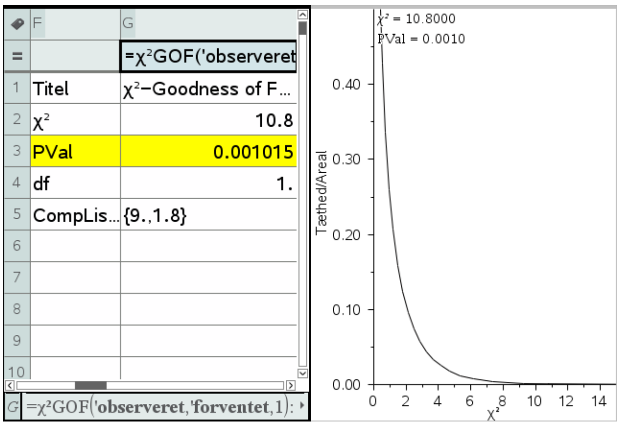
Vi slutter vores diskussion af terningekast med at vise hvordan hypotesetesten kan håndteres som en χ^2-test. I så fald skal vi tage hånd om *alle* de kategoriske variable, dels antallet af seksere (succeser), men også antallet af ikke-seksere (fiaskoer). Med to kategorier er der altså tale om én frihedsgrad. De forventede værdier har vi allerede gjort rede for:

Kategori	*Observeret værdi*	*Forventet værdi*
sekser	10	4
ikke-sekser	14	20

På basis af lister over de observerede værdier og de forventede værdier kan vi nu udføre en **Goodness-of-fit** test som vist:







Igen ligger p-værdien noget lavere end den eksakte værdi, men som før er det oplagt at nulhypotesen må forkastes: Den observerede fordeling ligger for langt fra den forventede forventede fordeling, til at nulhyptesen kan opretholde sin troværdighed.

Venstreorienterede piger

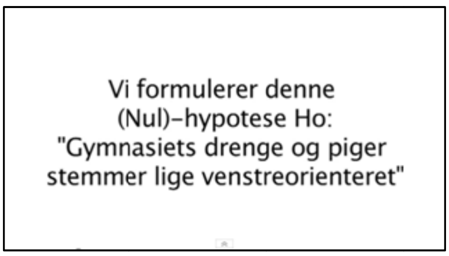
Som et eksempel på en uafhængighedstest vil vi nu se nærmere på en video 'Er piger venstreorienterede?' fra Gladsakse Gymnasium. Det er en kompleks problemstilling med mange fine detaljer, så det kan kun anbefales at se filmen i sin helhed i forbindelse med at man genemarbejder det følgende afsnit.

I filmen diskuterer de to hovedpersoner, Rasmus og Sidsel udfaldet af det kommende valg. Sidsel er overbevist om at rød blok vil vinde, ikke mindst fordi hun er overbevist om at kvinder er mere venstreorienterede end mænd. Det tror Rasumes imidlertid ikke på: Han regner ikke med at der er nogen sammenhæng mellem køn og den politiske holdning.

De bliver enige om at afgøre sagen ved at uddele spørgeskemaer til to klasser på Gladsakse gymnasium og se på om de adspurgte elever kan støtte Sidsels påstand om at piger er mere venstreorienterede. De er godt klar over, at det er problematisk hvorvidt en stikprøve beståeende af to klasser på Gladsakse Gymnasium kan siges at være repræsentativ for en population og i givet fald hvilken population. De bliver enige om at stikprøven kan siges at være repræsentativ *for populationen bestående af Storkøbenhavns gymnasieungdom*.

De bliver også enige om at *den røde blok* omfatter Socialdemokraterne, Socialistisk Folkepart, Enhedslisten og de Radikale, mens *den blå blok* omfatter resten af partierne.

De beslutter at teste nulhypotesen H_0, *ifølge hvilken en eventuelt forskel mellem de to køn alene skyldes de uundgåelige tilfældige udsving i en sådan stikprøve*, dvs. Rasmus hypotese. Sidsels hypotese repræsenterer til gengæld den alternative hypotese H_a, *at der rent faktisk er en systematisk forskel mellem holdningerne hos de to køn* (og mere præcist, at pigerne faktisk er mere venstreorienterede, men i første omgang altså blot, at den politiske holdning rent faktisk i et eller andet omfang afhænger af kønnet). Herefter går Sidsel ud og uddeler spørgeskemaerne hos de to klasser og vender en halv time senere tilbage med resultatet af undersøgelsen.



7: Hypotesetest

Der var 60 elever i skole i de to klasser bestående af 35 piger og 25 drenge. Stemmerne fordelte sig med 33 stemmer på den røde blok og 27 stemmer på den blå blok. *Endelig var der 25 røde piger*, dvs. af de 35 piger stemte de 25 på den røde blok.

Dette er vores observation, der nu skal bruges til at afgøre om vi kan forkaste nulhypotesen. De starter derfor med at udregne det forventede antal piger på baggrund af nulhypotesen. Ifølge nulhypotesen er der ingen forskel på de to køn og de følger derfor det samme mønster, ifølge hvilket 33 ud af 60 stemmer går til den røde blok, dvs. den røde blok udgør 33/60 af alle stemmerne. Da der er 35 piger vil det forventede antal røde piger derfor være

$$\frac{33}{60} \cdot 35 = 19.25$$

røde piger. Da det er et forventet antal, behøver det ikke at være et helt antal. Det angiver blot en *gennemsnitsværdi* for de røde piger, hvis der gentagne gange udtrækkes en tilfældig stikprøve bestående af 60 elever fra Storkøbenhavns Gymnasieungdom.

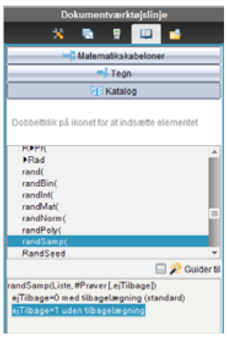
Men vi observerede 25 røde piger i vores stikprøve, så spørgsmålet er nu om det er tilstrækkeligt mange til at det ikke længere er rimeligt at forklare forskellen alene som et resultat af de tilfældige udsving i stikprøven, hvor nulhypotesen bør forkastes.

Simulering af nulhypotesen

For at afgøre det må vi undersøge, hvor nemt det er at finde 25 røde piger i en tilfældig stikprøve af samme slags som den foretagne. Rasmus og Sidsel bliver derfor enige om en simuleringstest, som de udfører meget snedigt ved hjælp af spillekort.

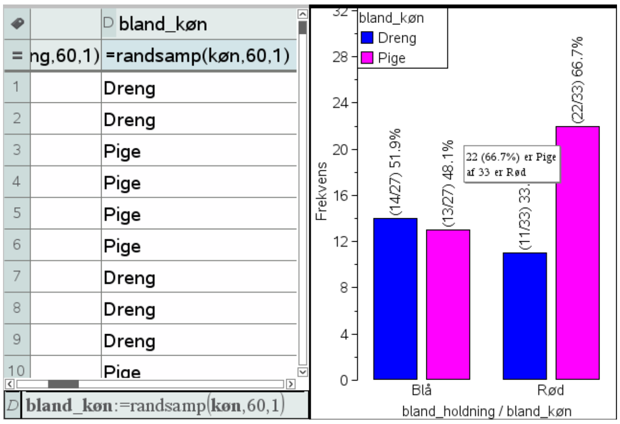
Rasmus får udleveret et sæt spillekort med 33 røde kort og 27 sorte kort, der repræsenterer elevernes holdning med røde kort hørende til den røde blok og sorte kort hørende til den blå blok. Sidsel får tilsvarende udleveret 35 røde kort og 25 sorte kort, der repræsenterer elevernes køn med de røde kort som piger og de sorte kort som drenge.

De blander nu kortene grundigt, så enhver korrespondance mellem køn og holdnng bliver brudt, dvs. de blandede kort repræsenterer nulhypotesen. De skiftes da til at lægge et kort frem for sig: Hvis begge kortene er røde svarer det til en pige, der stemmer på rød blok. Når de er færdige med at lægge de 60 kort frem foran sig har de samtidigt talt, hvor mange gange der blev lagt røde kort frem fra begge sider, dvs. hvor mange røde piger, der var i den tilfældige udtrækning. Det er denne simulering af nulhypotesen vi nu flytter over på comptueren ☺



Vi starter derfor med at åbne et **Lister og regneark**-værksted og oprette to lister, en for **holdning** bestående af 33 celler med teksten "Rød" og 27 celler med teksten "Blå", og en for **køn** med 35 celler med teksten "Pige" og 25 celler med teksten "Dreng". Disse to lister repræsenterer nu de to kortspil, som udleveres til Rasmus henholdsvis Sidsel. Vi blander derefter kortspillene godt og grundigt ved hjælp af en **RandSamp**-kommando, hvor vi denne gang trækker *uden tilbagelægning*, dvs. de blandede lister består af præcis de samme kort, men denne gang i en tilfældig rækkefølge:

Vi opretter et **grupperet søjlediagram** i et **Diagram og Statistik**-vindue (først afsættes **bland_holdning**, derefter højreklikkes i *x*-aksefeltet og der vælges **Opdel kategori efter variabel**, her **bland_køn**). I dette tilfælde fangede vi altså 22 røde piger.



Desværre kan vi ikke lægge en overligger ind på 25, fordi der ikke er tale om histogram men et søjlediagram. Så man skal være lidt mere omhyggelig for at tjekke om et udfald er skævt, dvs. fanger mindst 25 røde piger.

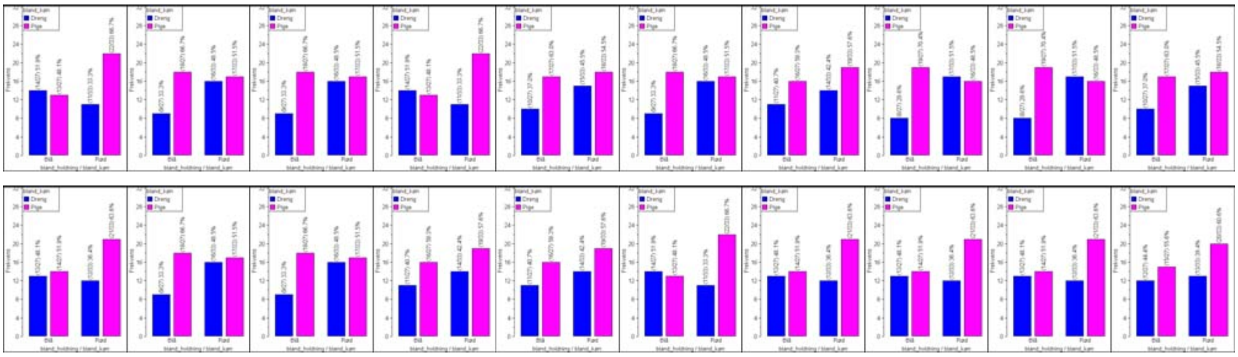
Proto-test og signifikansniveau

Vi er nu næsten klar til at udføre en første primitiv test, en *proto-test*, for at teste nulhypotesens troværdighed. Men før vi udfører proto-testen skal vi igen vælge et signifikansniveau. Vi holder fast i de 5%.

Sandsynligheden for at få et *skævt udfald* (mindst 25 røde piger), når man udspørger 60 elever med en tilfældig kobling mellem holdning og køn (nulhypotesen) kaldes p-værdien.

Hvis p-værdien falder under signifikansniveauet på 5%, dvs. hvis de skæve udfald forekommer i færre end hver tyvende prøverunde, regnes nulhypotesen for så utroværdig, at den forkastes. For at danne os et skøn over p-værdien kan vi derfor starte med at udføre simuleringen 20 gange. Hvis der ikke forekommer et eneste skævt udfald, tyder det på at p-værdien ligger under 5%, hvorfor nulhypotesen må forkastes. Hvis der omvendt forekommer et eller flere skæve udfald, tyder det på at p-værdien ligger over 5%, hvorfor vi ikke kan forkaste nulhypotesen, der derfor ind til videre står til troende ☺

Vi gentager derfor simuleringen 20 gange (ved at taste **CTRL/CMD R**, **CTRL/CMD .**) og tjekker om vi 'rammer overliggeren' for de røde piger ☺



Som det ses, lykkedes det ikke en eneste gang for de røde piger at ramme 'overliggeren'. Der var altså ingen skæve udfald i vores proto-test. Skønnet for p-værdien er derfor, at p-værdien ligger *under* 5%, hvorfor det ser ud til at vi må forkaste nulhypotesen.

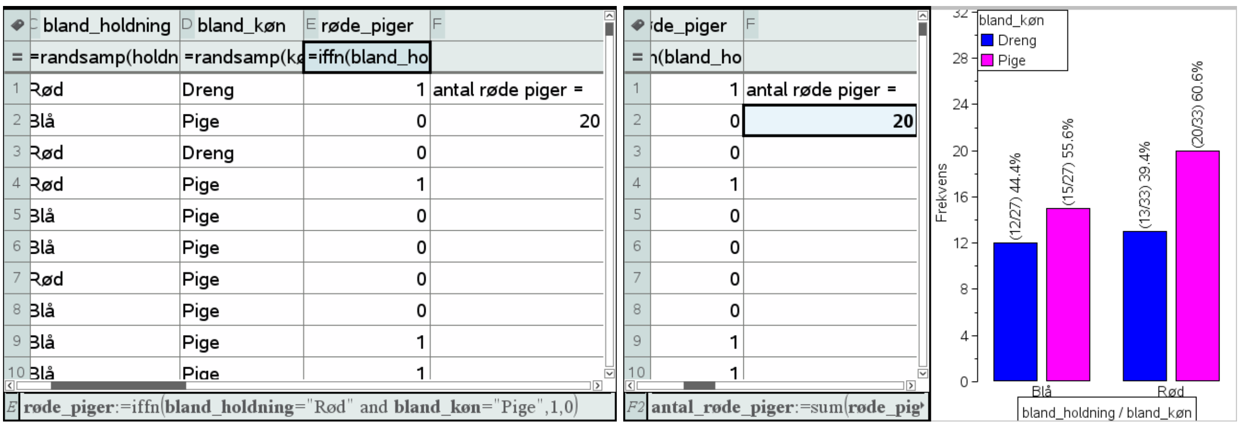
Men 20 simuleringer er selvfølgelig et spinkelt grundlag at drage vidtrækkende konklusioner på ☺. Så vi ser nu på, hvordan vi kan sætte antallet af simuleringer i vejret på en systematisk måde.

Testværdi og datafangst med LUA-kommandoen *capture*

Vi har hidtil forladt os på en visuel afgørelse af om et udfald er skævt. Vi vil nu også *regne* på sagen: Vi skal da først indføre en formel, der tæller antallet af røde piger, som er den *testværdi*, der skal bruges til at afgøre om en simulering giver anledning til et *skævt udfald*, dvs. et udfald, hvor antallet af røde piger er mindst 25. Da vi skal tælle *samtidigt* i to lister, skal vi være lidt mere omhyggelige denne gang. Det er *ikke* nok at bruge en **countIf**-kommando. I stedet opretter vi en liste **røde_piger**, der ved hjælp af **ifFn**-kommandoen

$$\textbf{røde\_piger} := \text{ifFn}('\textbf{bland\_holdning} = \text{"Rød"} \text{ and } '\textbf{bland\_køn} = \text{"Pige"},1,0)$$

tjekker række for række, om der står "Rød" i den første liste og "Pige" i den anden liste. I givet fald noteres et 1-tal (gevinst), ellers noteres et 0-tal (nitte). Antallet af røde piger er da netop summen af denne liste. Som det ses er der overensstemmelse mellem optællingen som beregning og den grafiske optælling.



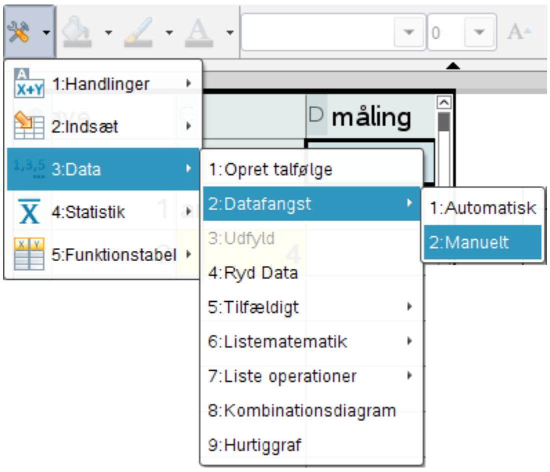
Vi kan nu opsamle antallet af røde piger ved at gemme dem som en variabel, dvs. vi omskriver celleformlen til

$$\boxed{F2}\ \textbf{antal\_røde\_piger} := \text{sum}('\textbf{røde\_piger})$$

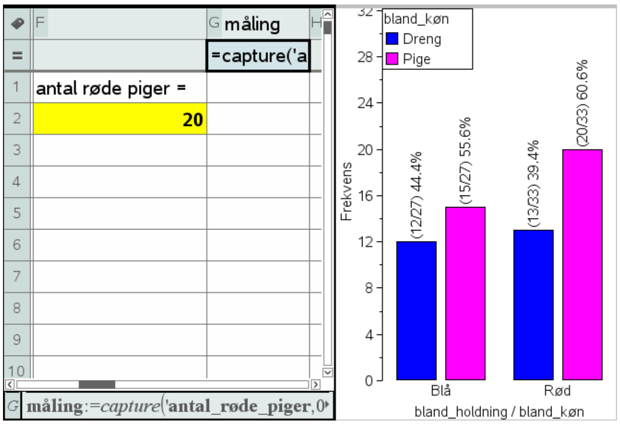
Antal røde piger er nu lagret som en variabel. Læg specielt mærke til, at tallet nu skrives i fed for at markere, at det er lagret som en variabel.

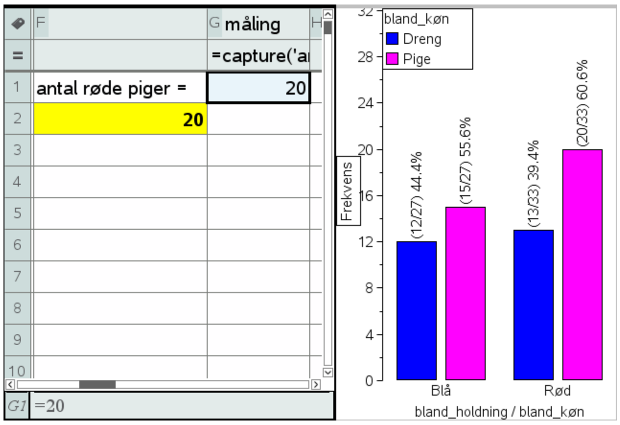
Obs
En **datafangst** med capture-kommandoen overvåger en variabel i variabelregistret. En **automatisk** datafangst registrerer hver gang variabel skifter værdi. En **manuel** datafangst registrerer variablens værdi når datafangsten udløses med **CTRL/CMD punktum**.

Vi kan derfor udføre en datafangst ved hjælp af LUA-kommandoen *capture*. Det er samme teknik, som vi allerede har beskrevet i kapitel 6 om sandsynlighedsregning. Vi opretter derfor en ny søjle kaldet **måling**, og vælger den følgende kommando i **Data**-menuen



Her vælger vi en **Manuel datafangst**, da antallet af røde piger jo ikke nødvendigvis skifter værdi, bare fordi vi udfører en ny simulering. Resultatet er den følgende formel





Obs
Capture-kommandoen virker kun i **Lister og regneark**-værkstedet. Det er jo netop *ikke* en matematik-kommando, men en LUA-kommando. Derimod virker udløsningskommandoen **CTRL/CMD .** fra ethvert værksted.

Læg mærke til, at *capture*-kommandoen står i kursiv – det er netop *ikke* en matematik-kommando, men en LUA-kommando. Inde i denne *capture*-kommando skal du nu angive hvilken variabel vi skal fange, her variablen **antal_røde_piger**. Endelig er der en parameter 0, der fortæller LUA-proceduren, at der er tale om en mauel datafangst.

Læg også mærke til, at der i første omgang slet ikke sker noget! Men det er jo fordi der er tale om en manuel datafangst, dvs. vi skal selv udløse datafangsten. Det sker ved at taste **CTRL/CMD .** (Kontrol Punktum). Det er ligegyldigt, hvor vi befinder os, når vi taster **CTRL/CMD .** – den manuelle datafangst udløses altid af denne taste-kombination, også selvom vi befinder os i et helt andet værksted.

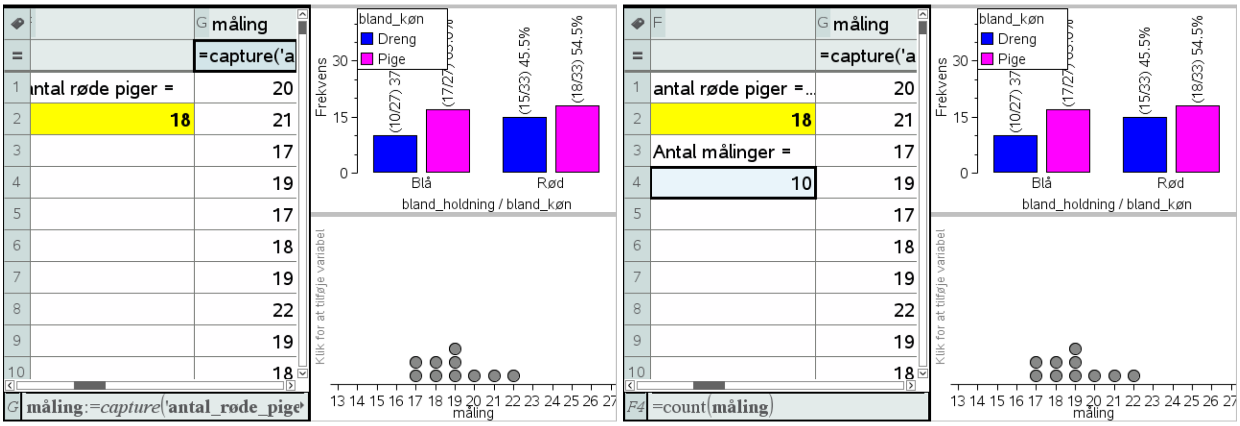
Nu skete der så endeligt noget! Vi har fanget antallet af rød piger i første simulering ☺

Obs
Husk at stå i **Lister og regneark**-værkstedet, for det er kun her at **CTRL/CMD R** udløser en genberegning af regnearket og dermed en ny simulering.

Så skal vi holde tungen lige i munden. Vi skal nu stille os i **Lister og regneark**-værkstedet og skiftetvis taste **CTRL/CMD R** for at udføre en ny simulering henholdsvis **CTRL/CMD .** for at udløse en ny datafangst!

Det kan godt lyde lidt indviklet men med lidt øvelse kan man fx holde **CTRL/CMD**-tasten ned med venstre hånd, mens man med højre hånd skiftevis taster **R** henholdsvis **Punktum**.

Resultatet af de første 10 simuleringer kan fx se således ud:



Inden vi går i gang med at lave datafangst i større omfang vil vi lige klargøre datafangsten, så vi dels kan se hvor mange målinger vi har foretaget, dels kan følge opbygningen af fordelingen for disse målinger.

Antallet af målinger fanger vi med celleformlen

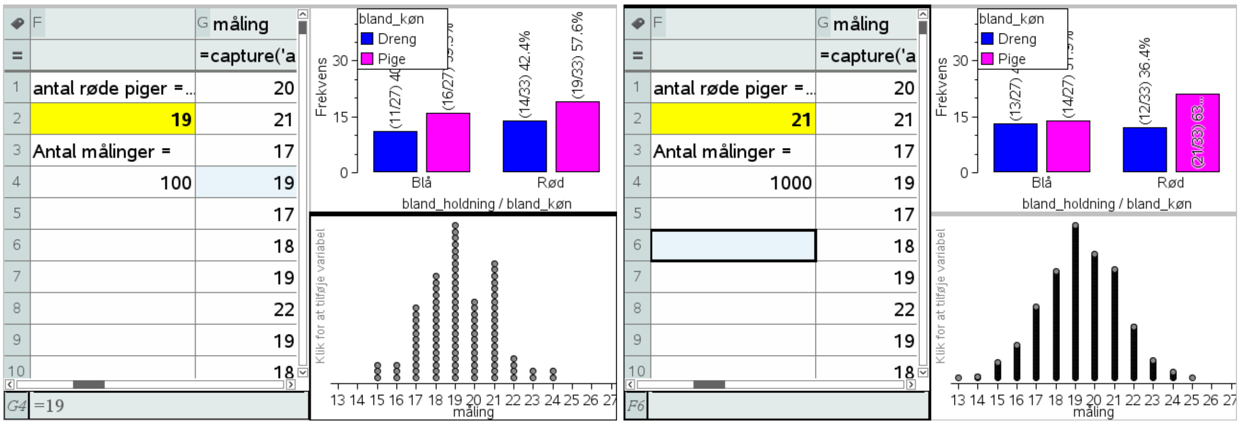
$$\boxed{F4} = \text{count}('\textbf{måling})$$

Visualiseringen af fordelingen klarer vi ved at oprette et nyt **Diagram og statistik**-vindue diagramtype til et prikdiagram for variablen **måling**.

Vi er så klar til at danse med fingrene, idet vi først taster **CTRL/CMD R** og derefter taster **CTRL/CMD punktum**.

Vi viser først resultatet af de første 100 målinger, derefter resultatet af de første 1000 målinger.

Der tegner sig da et mønster, hvor det hyppigste udfald som forventet er 19 røde piger. Men der er også en hel del udfald med fra 15 til 24 røde piger og faktisk også et enkelt skævt udfald med 25 røde piger (og tilsvarende et enkelt med 13 røde piger i den anden hale).

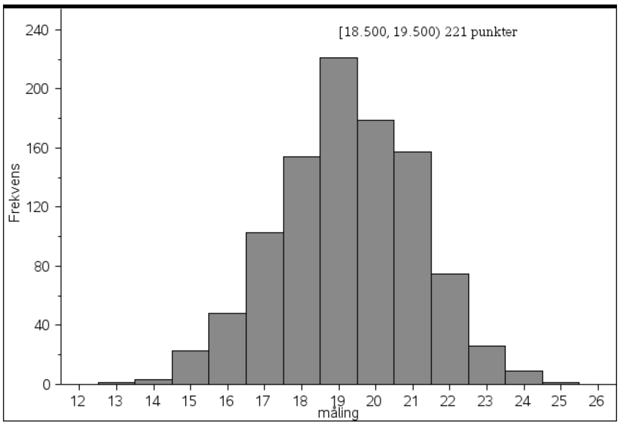


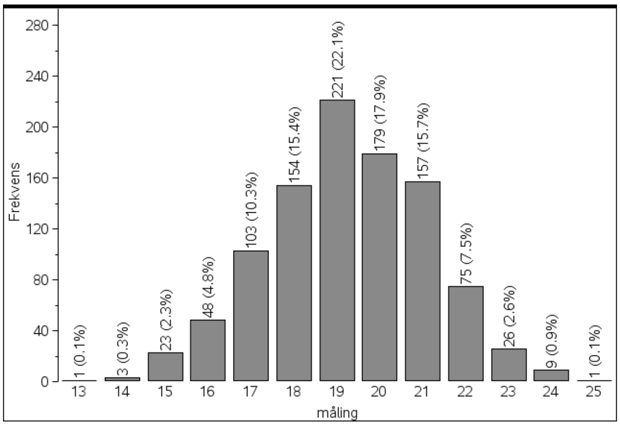
Vi kan danne os et bedre indtryk af fordelingen ved at skifte diagramtype til et histogram eller et søjlediagram (hvor vi så må tvinge målingen til at være en kategorisk variabel). I det sidste tilfælde kan vi så også slå **Vis alle etiketter** til.

I vores tilfælde ser vi da, at et skøn over p-værdien er givet ved

$$p \approx 1/1000 = 0.1\%$$

Sidsel får altså ret: Vi er nødt til at forkaste nulhypotesen om at der ingen sammenhæng er mellem politisk holdning og vælgerens køn.





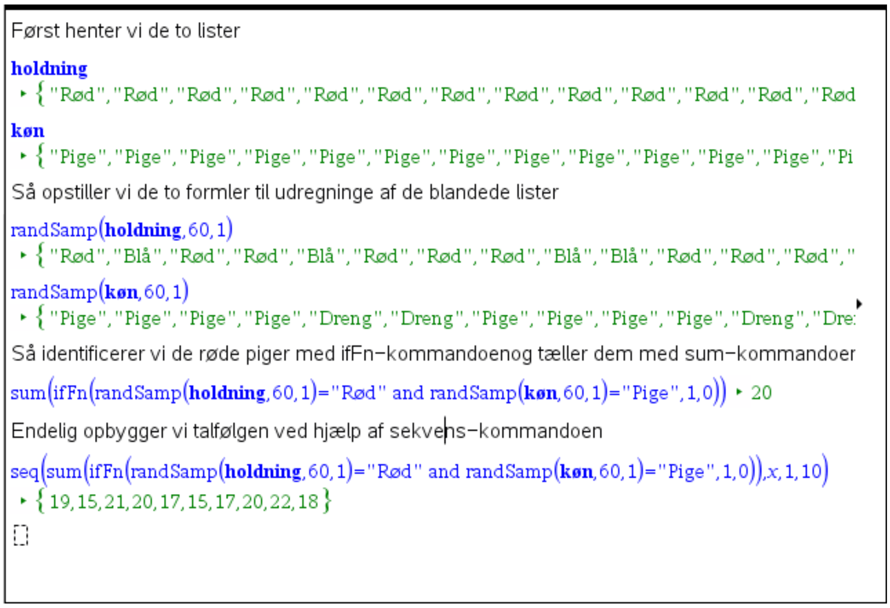
Det ville være rart hvis man kunne flytte målingen over på en automatisk datafangst, der er meget nemmere at udføre i praksis. Vi har da det problem, at hvis målingen gentager sin værdi, registreres den *ikke* automatisk. Der findes snedige tricks til at omgå dette problem, men de er netop snedige og vi vil derfor ikke gennemgå dem i denne eksempelsamlingen. Men der findes fx et hæfte om **Statistik med TI-Nspire CAS**, hvor man kan finde detaljerne.

Avanceret datafangst med sekvens-kommandoer*

Vi vil nu se på mulighederne for at udføre datafangsten i ét hug, dvs. udregne antallet af røde piger for 1000 simuleringer og opbygge en liste med de fundne værdier i ét hug, i stedet for trinvis, som vi så det med LUA-kommandoen *capture*. Hertil bruges sekvenskommandoen, der frembringer en talfølge ud fra et udtryk

$$\mathrm{seq}(udtryk, indeks, \mathrm{startværdi}, \mathrm{slutværdi})$$

Vi går frem ligesom før: Først oprettede vi lister over vælgernes holdning og køn. Dernæst omrørte vi listerne ved hjælp af RandSamp-kommandoen (dvs. vi blandede kortene, så enhver sammenhæng mellem holdning og køn i de to lister bliver brudt). Denne kommando er helt central i sekvenskommandoen, da det er den, der styrer dynamikken og skal sikre, at vi får et nyt antal seksere i hver simulering. Endelig benyttede vi en sum(ifFN)-kommando til at finde antallet af rød piger i de blandede lister. I første omgang tester vi nu vores sekvens-kommando ved blot at opsamle 10 værdier:



Som det kan ses virker **seq**-kommandoen! Hvis man indskriver den direkte i ét hug inde i fx et **Lister og regneark**-værksted, skal man være meget forsigtig med parenteserne, da det udtryk, der frembringer talfølgen, er sammensat af flere lag:

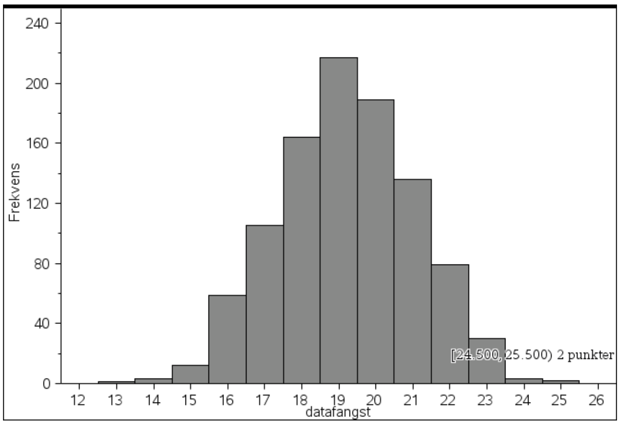
$$\mathrm{seq}\left(\boxed{\mathrm{sum}\left(\boxed{\mathrm{ifFn}\left(\boxed{\mathrm{randSamp}(\textbf{holdning},10,1)=\text{"Rød"}\text{ and }\mathrm{randSamp}(\textbf{køn},60,1)=\text{"Pige"},1,0}\right)}\right)},x,1,10\right)$$

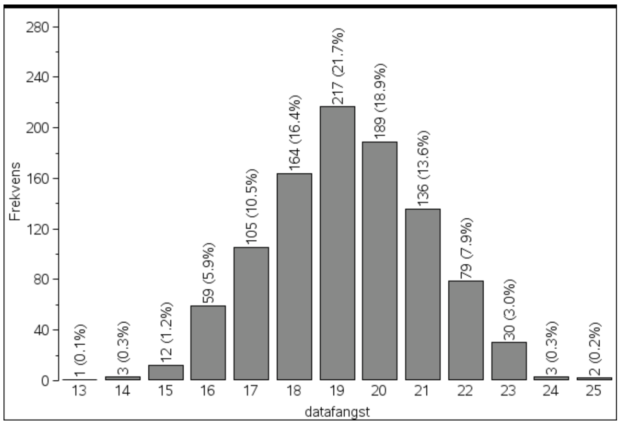
De yderste lag består af tælle-kommandoen, mens det inderste lag består af stikprøve-kommandoen.

Men når først det virker kan vi jo roligt navngive listen **datafangst** og sætte antallet af gentagelser op til 1000:

```
datafangst:=seq(countIf(randSamp(population,24),6),x,1,1000)
▸ {4,4,4,3,3,1,2,3,3,4,4,3,2,5,3,4,1,7,5,9,2,5,2,6,7,4,9,4,2,4,2,3,4,5,5,2,1,0,4,5,5,6,4,4,5,1,6,6,4
```

Denne gang går der et lille øjeblik før vi har hentet de 1000 målinger! For at kunne overskue dem kan vi derefter afbilde dem i et **Diagrammer og Statistik**-værksted





I begge tilfælde ser vi da igen, at sandsynligheden for at få et skævt udfald er langt under 5%. Denne gang er vores skøn over p-værdien baseret på 1000 gentagne simu-leringer givet ved 0.2%.

Sandsynligheden for at finde mindst 25 røde piger synes altså stadig at ligge langt under signifikansniveauet, hvorfor nulhypotesen forkastes som utroværdig og Sidsel får ret!

Man kan synes det er lidt kompliceret med de indpakkede sekvenskommandoer. Hvis man kender lidt til programmering er det da også muligt at oprette en brugerdefineret funktion, med en langt mere overskuelig opbygning. Men i dette hæfte vil vi ikke komme nærmere ind på programmering med TI-Nspire CAS. Det kan man i stedet læse mere om i et særskilt hæfte om **Programmering med TI-Nspire CAS**.

Uafhængighed som χ^2-test

Vi slutter vores diskussion af de venstreorenterede piger med at vise hvordan hypotese-testen kan håndteres som et χ^2-test. Vi skal da have adgang til en samlet tabel over de observerede værdier. Vi skal altså ikke blot tælle de røde piger, men også de blå piger og tilsvarende for de røde drenge og de blå drenge. Da vi kender det samlede antal drenge/piger og tilsvarende de samlede antal stemmer på rød blok/blå blok er det imidlertid nemt at finde disse manglende værdier. Først viser vi tabellen over de oplyste værdier. Ved at udnytte kendskab til summerne kan vi nu nemt udfylde resten (tabellen har kun én frihedsgrad, da antallet af røde piger fastlægger resten!)

	A	B	C	D	E	F
=						
1	Holdning\Køn	Pige	Dreng	I alt		
2	Rød	25		33		
3	Blå			27		
4	I alt	35	25	60		
5						
6						
7						
8						
9						
10						

B7

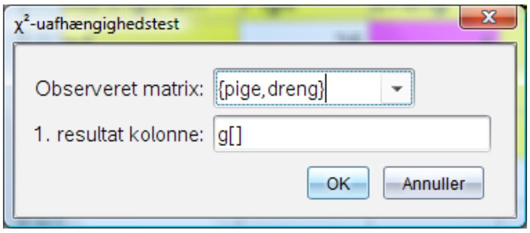
	A	B	C	D	E	F
=						
1	Holdning\Køn	Pige	Dreng	I alt		
2	Rød	25	8	33		
3	Blå	10	17	27		
4	I alt	35	25	60		
5						
6						
7						
8						
9						
10						

C6

Dernæst viser vi den afsluttede tabel. Til sidst flytter vi tabellen op over stregen, så søjlerne bliver navngivene variable.

		B	C	D	E pige	F dreng
=					=b2:b3	=c2:c3
1	oldning\Køn	Pige	Dreng	I alt	25	8
2	ød	25	8	33	10	17
3	å	10	17	27		
4	alt	35	25	60		
5						
6						
7						
8						
9						
10						

F4



På basis af tabellen over de observerede værdier kan vi nu udføre en **uafhængigheds-** test som vist. Denne gang får vi dog ikke tilbudt en grafisk illustration af testen:

		E pige	F dreng	G	H
=		=b2:b3	=c2:c3		=χ²2way({pige,dreng}): C
1		25	8	Titel	χ²−uafhængighedstest
2	33	10	17	χ²	9.16017
3	27			PVal	0.002473
4	60			df	1.
5				ExpMatrix	[[19.25,15.75][13.75,1...
6				CompMatrix	[[1.7175324675325,2.0...
7					
8					
9					
10					

H =χ²2way({pige,dreng}): CopyVar *Stat.*, *Stat1*.

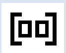
Igen ligger p-værdien langt lavere end signifikansværdien, hvorfor det er oplagt at nulhypotesen må forkastes: Den observerede fordeling ligger for langt fra den forventede forventede fordeling, til at nulhyptesen kan opretholde sin troværdighed. Rasmus taber og Sidsel vinder ☺

8 Vektorer og rumgeometri

TI-Nspire CAS håndterer vektorer via deres *koordinater* og understøtter tre forskellige måder at notere vektorer på: Som *liste-vektorer* og som *matrix*-vektorer, hvor matrix-vektorerne igen findes i to typer: som *række-vektorer* og som *søjle-vektorer*. Desværre håndteres de ikke fuldstændigt parallelt, så man er nødt til at træffe et valg om hvilken type, der skal være den foretrukne.

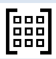
Tip
På en MAC kan man fx hente krøllede og kantede parenteser i tegnoversigten i venstre sidepanel.

Tip
Der findes skabeloner for 2-dimensionale rækkevektorer



og 2-dimensionale søjlevektorer

Men 3-dimensionale række- og søjle-vektorer kræver tabelskabelonen



Vektornotationer: Liste-, række- og søjle-vektorer

Traditionelt indskrives liste-vektorer med *krøllede parenteser*, fx

$$\mathbf{u}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$$

Tilsvarende skrives række-vektorer med *kantede parenteser*, hvor koordinaterne adskilles af *komma*, dvs.

$$v:=[1,2,3]$$

der omformes til

$$\mathbf{v}:=\begin{bmatrix}1 & 2 & 3\end{bmatrix} \blacktriangleright \begin{bmatrix}1 & 2 & 3\end{bmatrix}$$

Læg mærke til, at så snart man har tastet **ENTER** forsvinder kommaerne.
Endelig skrives søjlevektorer med *kantede parenteser*, hvor koordinaterne adskilles af semikolon, dvs.

$$w:=[1;2;3]$$

der omformes til

$$\mathbf{w}:=\begin{bmatrix}1\\2\\3\end{bmatrix} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix}$$

Læg igen mærke til, at så snart man har tastet **ENTER** forsvinder semikolonnerne.

Regning med vektorer

Obs
I den enkelte klasse kan man selvfølgelig træffe et andet valg, men man skal da være opmærksom på at enkelte formler i det følgende skal skrives på en lidt anden måde, dvs. resten af kapitlet skal i givet fald 'transponeres' til den anden vektortype.

Vi vil nu se nærmere på hvordan TI-Nspire CAS håndterer regning med vektorer. Vi er da nødt til at træffe et valg, da vektorer jo kan repræsenteres på forskellig vis som *liste-vektorer*, *række-vektorer* eller som *søjle-vektorer*. Hvis man bare vil lære at udføre nogle simple rutineberegninger med vektorer er det ligegyldigt hvilken af disse man anvender. Der er da fx en udbredt tradition for at anvende søjlevektorer. Men om man skriver vektorers koordinater vandret eller lodret er selvfølgelig blot en konvention uden egentligt matematisk indhold.

I samme øjeblik man ønsker at håndtere vektorer dynamisk, interaktivt og i forskellige værksteder er der imidlertid stor forskel, idet *listevektorer er langt mere fleksible at arbejde med end matrix-vektorer, dvs. række- og søjle-vektorer*. Vi har derfor valgt i det følgende at gennemgå brugen af liste-vektorer. Under alle omstændigheder er det *ikke* anbefalelsesværdigt at sammenblande de tre typer i undervisningen, så man sommetider anvender den ene, sommetider den anden ☺

I TI-Nspire CAS skelner man *ikke* mellem punkter og vektorer, dvs. man regner på punkter, præcis som man regner med vektorer. Teknisk set arbejder man altså indenfor en Euklidisk plan med et udvalgt begyndelsespunkt, Origo O, hvorfor man kan identificere et punkt P med dets stedvektor $\overrightarrow{OP}$. Vi kan derfor tillade os at skrive forbindelsesvektoren mellem to punkter P og Q således

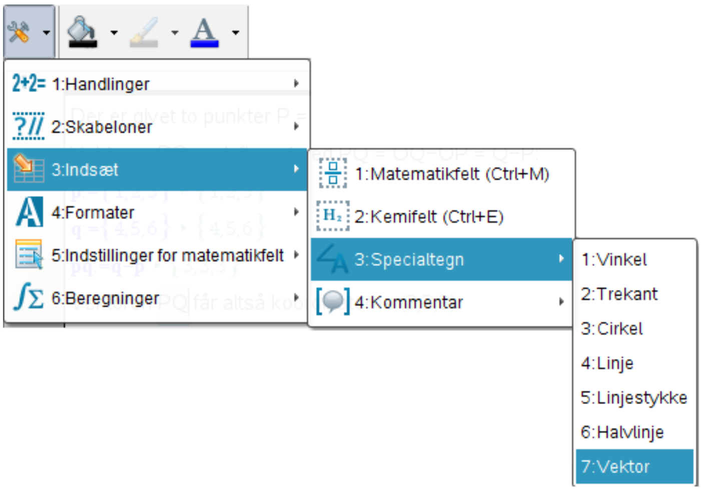
$$\overrightarrow{PQ}=\overrightarrow{OQ}-\overrightarrow{OP}=Q-P$$

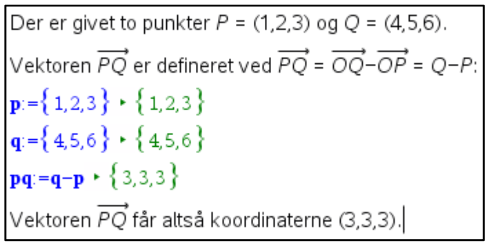
Og det er sådan vi vil regne med vektorer defineret ud fra punkter.

8: Vektorer og rumgeometri

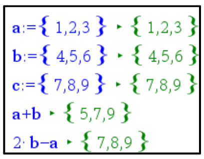
Vi skal nu være opmærksomme på, at TI-Nspire CAS *ikke* understøtter vektorpile inde i matematikfelterne. Men man kan sagtens sætte vektorpile udenfor matematikfelterne i den almindelige tekst. Hertil bruges vektorskabelonen fra værktøjsmenuen **Indsæt**

3: Indsæt ▸ 3: Specialtegn ▸ 7: Vektor





Da TI-Nspire CAS heller *ikke* skelner mellem små og store bogstaver er det altså i den ledsagende tekst, det tydeliggøres, hvad der er punkter, og hvad der er vektorer. Selvfølgelig er der nogle konventioner, hvor fx bogstaverne *P* og *Q* traditionelt står for punkter, mens bogstaverne *u*, *v* og *w* traditionelt står for vektorer (og som vi skal se senere, er vi ikke altid glade for bogstavet *u*, der er reserveret til parameterfremstillinger i rummet). Men i mange sammenhænge er det naturligt at bruge fx bogstaverne *a*, *b*, *c* og *d*, og så er det knap så klart om der er tale om punkter eller vektorer. Så fx bogstavet **a** kan både stå for punktet *A* og vektoren $\vec{a}$. Man kan da fx udvide navnet, så det er klart man har med en vektor at gøre, fx ved at kalde vektoren for **vektor_a**.



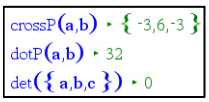
Med disse forhold bragt på plads kan vi nu se på, hvordan TI-Nspire CAS understøtter regneoperationer for vektorer.

Det er helt trivielt at lægge vektorer sammen eller trække dem fra hinanden.

Det er også trivielt at gange en vektor med et tal (skalar).

A a	B b	C plus	D linkomb
		='a+'b	=2*'b−'a
1	4	5	7
2	5	7	8
3	6	9	9

Når det kommer til produkter med vektorer er det mere kompliceret: De to typer produkter, skalarproduktet $\vec{a} \bullet \vec{b}$ og vektorproduktet $\vec{a} \times \vec{b}$, repræsenteres af specialkommandoerne DotP for prikprodukt (dvs. skalarproduktet) og CrossP for krydsproduktet (dvs. vektorproduktet).



Dertil kommer diverse matrix-operationer, såsom længder, determinanter og lignende. Her skal man være opmærksom på at matrix-operationer ikke umiddelbart understøtter liste-vektorer. Disse må først omdannes til matrix-vektorer. Det er heldigvis meget simpelt, idet en matrix *ikke* er andet end en liste af lister.

E kryds	F
=crossp('a,'b)	
−3	dotP(a,b)=
6	32
−3	det({a,b,c})=
	0

Man omdanner derfor en liste-vektor til en matrix-vektor ved at sætte et ekstra sæt krøllede parenteser, dvs. ved at skrive {**a**} i stedet for blot **a**. Tilsvarende omdannes et par af vektorer **a**,**b** til en matrix ved at sætte krøllede parenteser uden om, dvs. ved at skrive {**a**,**b**} i stedet for blot **a**,**b** osv.

Regneoperationerne kan bekvemt samles i følgende skema

8: Vektorer og rumgeometri

Her er de grønne regneoperationer vektor-kommandoer (der virker på alle tre repræsentationer af vektorerne).

De blå regneoperationer er matrix-kommandoer tillempet liste-vektorer (og som de står i skemaet virker de altså *kun* på liste-vektorer, der efterfølgende er pakket ind i krøllede parenteser, så de fremstår som matricer).

Endelig er de røde regneoperationer anført som formler, da TI-Nspire CAS ikke understøtter disse regneoperationer med indbyggede kommandoer.

Beskrivelse	Notation	Type	Kommentar
Summen af to vektorer $\vec{a}+\vec{b}$	$\mathbf{a}+\mathbf{b}=$ $\{a_1+b_1, a_2+b_2, a_3+b_3\}$	Vektor	
Differensen af to vektorer $\vec{a}-\vec{b}$	$\mathbf{a}-\mathbf{b}=$ $\{a_1-b_1, a_2-b_2, a_3-b_3\}$	Vektor	
Produktet af en vektor med en skalar $k\cdot\vec{a}$	$k\cdot\mathbf{a}=$ $\{k\cdot a_1, k\cdot a_2, k\cdot a_3\}$	Vektor	
Skalarproduktet af to vektorer $\vec{a}\bullet\vec{b}$	$\mathrm{dotP}(\mathbf{a},\mathbf{b})=$ $a_1\cdot b_1+a_2\cdot b_2+a_3\cdot b_3$	Skalar	
Vektorproduktet af to vektorer $\vec{a}\times\vec{b}$	$\mathrm{crossP}(\mathbf{a},\mathbf{b})=$ $\{a_3b_1-a_1b_3, a_2b_3-a_3b_2, a_1b_2-a_2b_1\}$	Vektor	I 2-dimensioner er vektorproduktet 3-dimensionalt!
Længden af en vektor $\lvert\vec{a}\rvert$	$\mathrm{norm}(\{\mathbf{a}\})=$ $\sqrt{a_1^2+a_2^2+a_3^2}$	Skalar	**Matrix-kommando**: Husk krøllet parentes!
Determinanten af to 2-dimensionale vektorer $\det(\vec{a},\vec{b})$	$\det(\{\mathbf{a},\mathbf{b}\})=$ $a_1\cdot b_2-a_2\cdot b_1$	Skalar	**Matrix-kommando**: Husk krøllet parentes!
Determinanten af tre 3-dimensionale vektorer $\det(\vec{a},\vec{b},\vec{c})$	$\det(\{\mathbf{a},\mathbf{b},\mathbf{c}\})=$ $a_1b_2c_3+a_2b_3c_1+a_3b_1c_2-$ $a_1b_3c_2-a_2b_1c_3-a_3b_2c_1$	Skalar	
Tværvektoren til en 2-d vektor $\hat{a}$	$\{-\mathbf{a}[2],\mathbf{a}[1]\}$	2d vektor	Findes *ikke* som kommando i TI-Nspire CAS!
Projektion af en vektor $\vec{a}$ på en vektor $\vec{b}$	$\dfrac{\mathrm{dotP}(\mathbf{a},\mathbf{b})}{\mathrm{dotP}(\mathbf{b},\mathbf{b})}\cdot\mathbf{b}$	Vektor	Findes *ikke* som kommando i TI-Nspire CAS!
Vinklen mellem to vektorer $\vec{a}$ og $\vec{b}$	$\cos^{-1}\left(\dfrac{\mathrm{dotP}(\mathbf{a},\mathbf{b})}{\mathrm{norm}(\{\mathbf{a}\})\cdot\mathrm{norm}(\{\mathbf{b}\})}\right)$	Vinkel	Findes *ikke* som kommando i TI-Nspire CAS!

A a	B b	C c
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3

D plus	E minus
='a+'b	='a−'b
a_1+b_1	a_1-b_1
a_2+b_2	a_2-b_2
a_3+b_3	a_3-b_3

F skalar	G produkt
	=skalar*'a
k	k*a_1
k	k*a_2
k	k*a_3

Skalarprodukt
dotP(a,b)
a_1*b_1+a_2*b_2+a_3*b_3

I krydsprodukt
=crossp('a,'b)
a_2*b_3−a_3*b_2
a_3*b_1−a_1*b_3
a_1*b_2−a_2*b_1

Længde
norm({a})
√ (a_1^2+a_2^2+a_3^2)

Determinant
det({a,b,c})
a_1*(b_2*c_3−b_3*c_2)−a_2*...

K tværvektor
={⁻'a[2],'a[1]}
⁻a_2
a_1

M projektion
=dotp('a,'b)/(dotp('b,'b))*'b
(a_1*b_1+a_2*b_2+a_3*b_3)*b_1/(b_1...
(a_1*b_1+a_2*b_2+a_3*b_3)*b_2/(b_1...
(a_1*b_1+a_2*b_2+a_3*b_3)*b_3/(b_1...

Regnearket er symbolsk ☺

Med regneoperationerne på plads er vi nu endelig klar til at løse opgaver ☺

Opgave 1: Statisk

I et koordinatsystem i rummet er der givet 3 vektorer

$$\vec{a}=\begin{pmatrix}1\\2\\3\end{pmatrix},\quad \vec{b}=\begin{pmatrix}2\\-1\\2\end{pmatrix},\quad \vec{c}=\begin{pmatrix}-1\\2\\3\end{pmatrix}$$

a) Bestem et gradtal for vinklen mellem $\vec{a}$ og $\vec{b}$.

b) Bestem koordinatsættet til projektionen af $\vec{a}$ på $\vec{c}$.

c) Bestem tallene s og t, således at vektoren

$$\vec{d}=\vec{a}+s\cdot\vec{b}+t\cdot\vec{c}$$

står vinkelret på både $\vec{b}$ og $\vec{c}$, og angiv koordinaterne for $\vec{d}$.

Opgaven løses i et **Note**-værksted. Det kan ske på flere måder. Den simpleste er at arbejde *statisk*, dvs. omhyggeligt kopiere og indsætte undervejs. Det kan fx se således ud, hvor vi har skiftet til fuld skærm og arbejder med noter i to spalter:

Opgave i vektorregning

Først defineres de tre vektorer $\vec{a}$, $\vec{b}$ og $\vec{c}$:

$\mathbf{a}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$ $\quad\mathbf{b}:=\{2,{}^-1,2\} \blacktriangleright \{2,{}^-1,2\}$

$\mathbf{c}:=\{{}^-1,2,3\} \blacktriangleright \{{}^-1,2,3\}$

a) Vinklen v mellem de to vektorer $\vec{a}$ og $\vec{b}$ findes (hvor der arbejdes i grader):

$$v:=\cos^{-1}\left(\frac{\mathrm{dotP}(\mathbf{a},\mathbf{b})}{\mathrm{norm}(\{\mathbf{a}\})\cdot\mathrm{norm}(\{\mathbf{b}\})}\right) \blacktriangleright 57.6885$$

Et gradtal for vinklen mellem de to vektorer $\vec{a}$ og $\vec{b}$ er altså givet ved $\boxed{57.7}$.

b) Projektionen $\vec{a}_c$ af vektoren $\vec{a}$ på vektoren $\vec{c}$ bestemmes:

$$\mathbf{a_c}:=\frac{\mathrm{dotP}(\mathbf{a},\mathbf{c})}{\mathrm{dotP}(\mathbf{c},\mathbf{c})}\cdot\mathbf{c} \blacktriangleright \left\{\frac{-6}{7},\frac{12}{7},\frac{18}{7}\right\}$$

Projektionen af vektoren $\vec{a}$ på vektoren $\vec{c}$ er altså givet ved $\boxed{\begin{bmatrix}\frac{-6}{7}\\ \frac{12}{7}\\ \frac{18}{7}\end{bmatrix}}$.

c) Først defineres vektoren $\vec{d}$:

$\mathbf{d}:=\mathbf{a}+s\cdot\mathbf{b}+t\cdot\mathbf{c} \blacktriangleright \{2\cdot s-t+1,{}^-s+2\cdot t+2,2\cdot s+3\cdot t+3\}$

Vektoren $\vec{d}$ skal stå vinkelret på såvel $\vec{b}$ som $\vec{c}$:

$\mathrm{dotP}(\mathbf{d},\mathbf{b})=0 \blacktriangleright 9\cdot s+2\cdot t+6=0$

$\mathrm{dotP}(\mathbf{d},\mathbf{c})=0 \blacktriangleright 2\cdot s+14\cdot t+12=0$

Vi skal altså løse ligningssystemet

$\boxed{9\cdot s+2\cdot t+6=0 \qquad 2\cdot s+14\cdot t+12=0}$

Det gøres ved solvekommandoen

$$\mathrm{solve}\left(\begin{cases}9\cdot s+2\cdot t+6=0\\2\cdot s+14\cdot t+12=0\end{cases},s,t\right) \blacktriangleright s=\frac{-30}{61} \text{ and } t=\frac{-48}{61}$$

Tallene s og t er altså givet ved $\boxed{s=\frac{-30}{61} \text{ og } t=\frac{-48}{61}}$.

Disse indsættes i udtrykket for vektoren $\vec{d}$:

$$\mathbf{d}|s=\frac{-30}{61} \text{ and } t=\frac{-48}{61} \blacktriangleright \left\{\frac{49}{61},\frac{56}{61},\frac{-21}{61}\right\}$$

Vektoren $\vec{d}$ har altså koordinaterne $\boxed{\begin{bmatrix}\frac{49}{61}\\ \frac{56}{61}\\ \frac{-21}{61}\end{bmatrix}}$.

Svagheden ved en statisk løsning er selvfølgelig at hvis vi har lavet en indtastningsfejl undervejs, skal vi tilbage og ikke bare rette indtastningsfejlen, men også kopiere og indsætte undervejs i hele besvarelsen, så den opdateres korrekt. På den ovenstående figur har vi med rødt markeret de steder, hvor man bare har kopieret og indsat eller hvor man selv skrevet svarene. Fx er søjlevektorerne skrevet direkte i 'hånden' i en skabelon.

Obs
En dynamisk interaktiv bevarelse kræver fortrolighed med vektorer som datastrukturer. Det kan du læse mere om i appendixet til dette kapitel.

Opgave 1: Interaktiv*

Det er langt mere udfordrende, men også mere hensigtsmæssigt at oprette besvarelsen så den er dynamisk interaktiv fra starten. Eventuelle indtastningsfejl kan da rettes trivielt hvorefter dokumentet selv retter til. På grund af kompleksiteten anbefaler vi dog, at man venter med at prøve kræfter med de dynamisk interaktive besvarelser til man er helt fortrolig med de statiske mere rutineprægede besvarelser.

I den venstre søjle skal man altså i stedet for selv at skrive 57.7 i konklusionen for vinklen skrive kommandoen round(**v**,1) og derefter skjule input. Tilsvarende skal man skrive kommandoen $\{\mathbf{v}\}^T$ i konklusionen for projektionen og derefter skjule input.

Det bliver straks mere interessant i anden søjle, hvor beregningen er mere kompliceret med mellemregninger undervejs, som vi skal sikre os *kæder korrekt*. Vi er derfor nødt til at navgive ligningerne **eq1** og **eq2**, så vi kan referere til dem i linSolve-kommandoen (linSolve er en forkortelse af lineær solve, dvs. den kan netop anvendes på lineære ligningssystemer).

Her har vi valgt linSolve-kommandoen i stedet for Solve-kommandoen, fordi linSolve afleverer svaret som en liste, der gør det nemt at referere til de enkelte løsninger for s og t.

Det er da vigtigt at gentage nogle af mellemregningerne undervejs, så kommandoerne ikke bliver for kompakte og svære at aflæse. Fx er linSolve-kommandoen nu meget kompakt, idet det kun fremgår, at man løser ligningerne **eq1** og **eq2**, så det er vigtigt at disse to ligninger præsenteres tydeligt i den ovenstående tekst.

Det kan fx se således ud:

Opgave i vektorregning

Først defineres de tre vektorer $\vec{a}$, $\vec{b}$ og $\vec{c}$

$\mathbf{a}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$ $\quad \mathbf{b}:=\{2,{}^-1,2\} \blacktriangleright \{2,{}^-1,2\}$

$\mathbf{c}:=\{{}^-1,2,3\} \blacktriangleright \{{}^-1,2,3\}$

a) Vinklen v mellem de to vektorer $\vec{a}$ og $\vec{b}$ findes (hvor der arbejdes i grader):

$$\mathbf{v}:=\cos^{-1}\left(\frac{\mathrm{dotP}(\mathbf{a},\mathbf{b})}{\mathrm{norm}(\{\mathbf{a}\})\cdot \mathrm{norm}(\{\mathbf{b}\})}\right) \blacktriangleright 57.6885$$

Et gradtal for vinklen mellem de to vektorer $\vec{a}$ og $\vec{b}$ er altså givet ved 57.7.

b) Projektionen $\vec{a}_c$ af vektoren $\vec{a}$ på vektoren $\vec{c}$ bestemmes:

$$\mathbf{a_c}:=\frac{\mathrm{dotP}(\mathbf{a},\mathbf{c})}{\mathrm{dotP}(\mathbf{c},\mathbf{c})}\cdot \mathbf{c} \blacktriangleright \left\{\frac{-6}{7},\frac{12}{7},\frac{18}{7}\right\}$$

Projektionen af vektoren $\vec{a}$ på vektoren $\vec{c}$ er altså givet ved $\begin{bmatrix}\frac{-6}{7}\\ \frac{12}{7}\\ \frac{18}{7}\end{bmatrix}$.

c) Først defineres vektoren $\vec{d}$:

$\mathbf{d}:=\mathbf{a}+s\cdot\mathbf{b}+t\cdot\mathbf{c} \blacktriangleright \{2\cdot s-t+1,{}^-s+2\cdot t+2,2\cdot s+3\cdot t+3\}$

Vektoren $\vec{d}$ skal stå vinkelret på såvel $\vec{b}$ som $\vec{c}$:

$\mathbf{eq1}:=\mathrm{dotP}(\mathbf{d},\mathbf{b})=0 \blacktriangleright 9\cdot s+2\cdot t+6=0$

$\mathbf{eq2}:=\mathrm{dotP}(\mathbf{d},\mathbf{c})=0 \blacktriangleright 2\cdot s+14\cdot t+12=0$

Vi skal altså løse det lineære ligningssystem

$9\cdot s+2\cdot t+6=0 \qquad 2\cdot s+14\cdot t+12=0$

Det gøres ved linSolvekommandoen

$$\mathbf{l\text{ø}sning}:=\mathrm{linSolve}\left(\begin{cases}\mathbf{eq1}\\ \mathbf{eq2}\end{cases},s,t\right) \blacktriangleright \left\{\frac{-30}{61},\frac{-48}{61}\right\}$$

Tallene s og t er altså givet ved $s=\frac{-30}{61}$ og $t=\frac{-48}{61}$.

Disse insættes i udtrykket for vektoren $\vec{d}$:

$\mathbf{d_{speciel}}:=\mathbf{d}|s=\mathbf{l\text{ø}sning}[1] \text{ and } t=\mathbf{l\text{ø}sning}[2]$

$\blacktriangleright \left\{\frac{49}{61},\frac{56}{61},\frac{-21}{61}\right\}$

Vektoren $\vec{d}$ har altså koordinaterne $\begin{bmatrix}\frac{49}{61}\\ \frac{56}{61}\\ \frac{-21}{61}\end{bmatrix}$

Her har vi med gult markeret alle de steder, hvor vi har skjult input. Foldes de ud, ser det således ud

Opgave i vektorregning

Først defineres de tre vektorer $\vec{a}$, $\vec{b}$ og $\vec{c}$

$\mathbf{a}:=\{1,2,3\}$ ▸ $\{1,2,3\}$ $\mathbf{b}:=\{2,-1,2\}$ ▸ $\{2,-1,2\}$

$\mathbf{c}:=\{-1,2,3\}$ ▸ $\{-1,2,3\}$

a) Vinklen v mellem de to vektorer $\vec{a}$ og $\vec{b}$ findes (hvor der arbejdes i grader):

$$v:=\cos^{-1}\left(\frac{\mathrm{dotP}(\mathbf{a},\mathbf{b})}{\mathrm{norm}(\{\mathbf{a}\})\cdot \mathrm{norm}(\{\mathbf{b}\})}\right) \blacktriangleright 57.6885$$

Et gradtal for vinklen mellem de to vektorer $\vec{a}$ og $\vec{b}$ er altså givet ved round(v,1) ▸ 57.7.

b) Projektionen $\vec{a}_c$ af vektoren $\vec{a}$ på vektoren $\vec{c}$ bestemmes:

$$\mathbf{a_c}:=\frac{\mathrm{dotP}(\mathbf{a},\mathbf{c})}{\mathrm{dotP}(\mathbf{c},\mathbf{c})}\cdot \mathbf{c} \blacktriangleright \left\{\frac{-6}{7},\frac{12}{7},\frac{18}{7}\right\}$$

Projektionen af vektoren $\vec{a}$ på vektoren $\vec{c}$ er altså givet ved $\{\mathbf{a_c}\}^T \blacktriangleright \begin{bmatrix}\frac{-6}{7}\\ \frac{12}{7}\\ \frac{18}{7}\end{bmatrix}$.

c) Først defineres vektoren $\vec{d}$:

$\mathbf{d}:=\mathbf{a}+s\cdot \mathbf{b}+t\cdot \mathbf{c}$ ▸ $\{2\cdot s-t+1, -s+2\cdot t+2, 2\cdot s+3\cdot t+3\}$

Vektoren $\vec{d}$ skal stå vinkelret på såvel $\vec{b}$ som $\vec{c}$:

$\mathbf{eq1}:=\mathrm{dotP}(\mathbf{d},\mathbf{b})=0$ ▸ $9\cdot s+2\cdot t+6=0$

$\mathbf{eq2}:=\mathrm{dotP}(\mathbf{d},\mathbf{c})=0$ ▸ $2\cdot s+14\cdot t+12=0$

Vi skal altså løse det lineære ligningssystem

$\mathbf{eq1}$ ▸ $9\cdot s+2\cdot t+6=0$ $\mathbf{eq2}$ ▸ $2\cdot s+14\cdot t+12=0$

Det gøres ved linSolvekommandoen

$$\mathbf{løsning}:=\mathrm{linSolve}\left(\begin{cases}\mathbf{eq1}\\ \mathbf{eq2}\end{cases}, s,t\right) \blacktriangleright \left\{\frac{-30}{61},\frac{-48}{61}\right\}$$

Tallene s og t er altså givet ved $s=\mathbf{løsning}[1] \blacktriangleright s=\frac{-30}{61}$ og $t=\mathbf{løsning}[2] \blacktriangleright t=\frac{-48}{61}$.

Disse insættes i udtrykket for vektoren $\vec{d}$:

$\mathbf{d_{speciel}}:=\mathbf{d}|s=\mathbf{løsning}[1]$ and $t=\mathbf{løsning}[2]$

▸ $\left\{\frac{49}{61},\frac{56}{61},\frac{-21}{61}\right\}$

Vektoren $\vec{d}$ har altså koordinaterne $(\{\mathbf{d_{speciel}}\})^T \blacktriangleright \begin{bmatrix}\frac{49}{61}\\ \frac{56}{61}\end{bmatrix}$

Man kan selvfølgelig altid diskutere, hvor langt man skal gå i interaktiviteten, men det sikreste er altså som vist også at lade konklusionerne være interaktive ☺.

Opgave 2: Statisk

I et koordinatsystem i rummet er givet et punkt $P(5,4,3)$. To linjer l og m er bestemt ved:

$$l:\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}8\\0\\0\end{pmatrix}+t\cdot\begin{pmatrix}1\\0\\1\end{pmatrix},\quad t\in R$$

$$m:\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}4\\-4\\2\end{pmatrix}+s\cdot\begin{pmatrix}1\\2\\0\end{pmatrix},\quad s\in R$$

a) Bestem en ligning for den plan α, der indeholder P og l.

b) Find koordinatsættet til m's skæringspunkt med α.

c) Bestem et gradtal for den spidse vinkel, som m danner med α.

d) Bestem parameterfremstillingen for den linje, der går gennem P og skærer både l og m.

Som før lægger vi ud med en statisk løsning, hvor vi omhyggeligt kopierer og indsætter udtryk undervejs eller skriver svarene direkte (markeret med rødt på figuren):

Opgave i vektorregning

Først defineres punktet P:

$p:=\{5,4,3\} \blacktriangleright \{5,4,3\}$

Dernæst oprettes parameterfremstillingerne for linjerne l og m ud fra ankerpunkterne P_1 og P_m og retningsvektorerne v_1 og v_m :

$p_1:=\{8,0,0\} \blacktriangleright \{8,0,0\}$ $\quad v_1:=\{1,0,1\} \blacktriangleright \{1,0,1\}$

$p_m:=\{4,-4,2\} \blacktriangleright \{4,-4,2\}$ $\quad v_m:=\{1,2,0\} \blacktriangleright \{1,2,0\}$

$l:=p_1+t\cdot v_1 \blacktriangleright \{t+8,0,t\}$ $\quad m:=p_m+s\cdot v_m \blacktriangleright \{s+4,2\cdot s-4,2\}$

a) Vi finder først endnu en retningsvektor $\overrightarrow{PP_l}$ for planen α:

$pp_1:=p_1-p \blacktriangleright \{3,-4,-3\}$

Ved at krydse den med retningsvektoren for linjen l fås nu normalvektoren $\vec{n}$ til planen α:

$n:=\text{crossP}(v_1,pp_1) \blacktriangleright \{4,6,-4\}$

Planens ligning findes ved indsættelse af normalvektorens koordinater $\mathbf{n}=(a,b,c)$ og ankerpunktets koordinater $P_1=(x_0,y_0,z_0)$ i ligningen

$a\cdot(x-x_0)+b\cdot(y-y_0)+c\cdot(z-z_0)=0$

$4\cdot(x-8)+6\cdot(y-0)-4\cdot(z-0)=0 \blacktriangleright 4\cdot x+6\cdot y-4\cdot z-32=0$

Planens ligning er altså givet ved $\boxed{4\cdot x+6\cdot y-4\cdot z-32=0}$.

b) Koordinatsættet til skæringspunktet Q mellem linjen m og planen α findes ved at løse ligningssystemet bestående af linjens parameterfremstilling og planens ligning

$$\text{solve}\left(\boxed{\begin{cases} x=s+4 \\ y=2\cdot s-4 \\ z=2 \\ 4\cdot x+6\cdot y-4\cdot z-32=0 \end{cases}},x,y,z,s\right)$$

$\blacktriangleright x=7 \text{ and } s=3 \text{ and } y=2 \text{ and } z=2$

Skæringspunktet Q har altså koordinaterne $\boxed{(7,2,2)}$

c) Vi finder gradtallet for vinklen u mellem retningsvektoren v_m for m og planens normalvektor $\vec{n}$ og trækker den fundne vinkel fra 90°:

$$u:=90-\cos^{-1}\left(\frac{\text{dotP}(v_m,n)}{\text{norm}(\{v_m\})\cdot\text{norm}(\{n\})}\right) \blacktriangleright 60.195$$

Den spidse vinkel mellem m og α er altså $\boxed{60.2°}$

d) Den linje k, der går gennem P og skærer både l og m, må gå gennem Q. Vi skal altså finde en parameterfremstilling for linjen gennem P og Q:

$k:=p+t\cdot(q-p) \blacktriangleright \{2\cdot t+5,4-2\cdot t,3-t\}$

En mulig parameterfremstilling er altså givet ved

$$\boxed{\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} 5 \\ 4 \\ 3 \end{bmatrix}+t\cdot\begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix}}.$$

Opgave 2: Interaktiv*

Opgave i vektorregning

Først defineres punktet P:

$p:=\{5,4,3\} \blacktriangleright \{5,4,3\}$

Dernæst oprettes parameterfremstillingerne for linjerne l og m ud fra ankerpunkterne P_1 og P_m og retningsvektorerne v_1 og v_m :

$p_1:=\{8,0,0\} \blacktriangleright \{8,0,0\}$ $\quad v_1:=\{1,0,1\} \blacktriangleright \{1,0,1\}$

$p_m:=\{4,-4,2\} \blacktriangleright \{4,-4,2\}$ $\quad v_m:=\{1,2,0\} \blacktriangleright \{1,2,0\}$

$l:=p_1+t\cdot v_1 \blacktriangleright \{t+8,0,t\}$ $\quad m:=p_m+s\cdot v_m \blacktriangleright \{s+4,2\cdot s-4,2\}$

a) Vi finder først endnu en retningsvektor $\overrightarrow{PP_l}$ for planen α:

$pp_1:=p_1-p \blacktriangleright \{3,-4,-3\}$

Ved at krydse den med retningsvektoren for linjen l fås nu normalvektoren $\vec{n}$ til planen α:

$n:=\text{crossP}(v_1,pp_1) \blacktriangleright \{4,6,-4\}$

Planens ligning findes

$eq\alpha:=\text{dotP}(\{x,y,z\}-p_1,n)=0 \blacktriangleright 4\cdot x+6\cdot y-4\cdot z-32=0$

Planens ligning er altså givet ved $4\cdot x+6\cdot y-4\cdot z-32=0$.

b) Koordinatsættet til skæringspunktet Q mellem linjen m og planen α findes ved at løse ligningssystemet bestående af linjens parameterfremstilling og planens ligning

$x=m[1] \blacktriangleright x=s+4$ $\quad y=m[2] \blacktriangleright y=2\cdot s-4$ $\quad z=m[3] \blacktriangleright z=2$

$$\text{løsning}:=\text{solve}\left(\begin{cases} x=m[1] \\ y=m[2] \\ z=m[3] \\ eq\alpha \end{cases},x,y,z,s\right)$$

$\blacktriangleright x=7 \text{ and } s=3 \text{ and } y=2 \text{ and } z=2$

$q:=m|\text{løsning} \blacktriangleright \{7,2,2\}$

Skæringspunktet Q har altså koordinaterne $\{7,2,2\}$.

c) Vi finder gradtallet for vinklen u mellem retningsvektoren v_m for m og planens normalvektor $\vec{n}$ og trækker den fundne vinkel fra 90°:

$$u:=90-\cos^{-1}\left(\frac{\text{dotP}(v_m,n)}{\text{norm}(\{v_m\})\cdot\text{norm}(\{n\})}\right) \blacktriangleright 60.195$$

Den spidse vinkel mellem m og α er altså 60.2.

d) Den linje k, der går gennem P og skærer både l og m, må gå gennem m's skæringspunkt Q med α. Vi skal altså finde en parameterfremstilling for linjen gennem P og Q:

$k:=p+t\cdot(q-p) \blacktriangleright \{2\cdot t+5,4-2\cdot t,3-t\}$

En mulig parameterfremstilling er altså givet ved

$$\begin{bmatrix} x=2\cdot t+5 \\ y=4-2\cdot t \\ z=3-t \end{bmatrix}$$

De skjulte kommandoer ser da fx således ud:

Opgave i vektorregning

Først defineres punktet *P*:

$\mathbf{p}:=\{5,4,3\} ▸ \{5,4,3\}$

Dernæst oprettes parameterfremstillingerne for linjerne *l* og *m* ud fra ankerpunkterne $\mathbf{P_l}$ og $\mathbf{P_m}$ og retningsvektorerne $\mathbf{v_l}$ og $\mathbf{v_m}$:

$\mathbf{p_l}:=\{8,0,0\} ▸ \{8,0,0\}$ $\quad \mathbf{v_l}:=\{1,0,1\} ▸ \{1,0,1\}$

$\mathbf{p_m}:=\{4,-4,2\} ▸ \{4,-4,2\}$ $\quad \mathbf{v_m}:=\{1,2,0\} ▸ \{1,2,0\}$

$\mathbf{l}:=\mathbf{p_l}+t\cdot \mathbf{v_l} ▸ \{t+8,0,t\}$ $\quad \mathbf{m}:=\mathbf{p_m}+s\cdot \mathbf{v_m} ▸ \{s+4,2\cdot s-4,2\}$

a) Vi finder først endnu en retningsvektor $\overrightarrow{PP_l}$ for planen α:

$\mathbf{pp_l}:=\mathbf{p_l}-\mathbf{p} ▸ \{3,-4,-3\}$

Ved at krydse den med retningsvektoren for linjen *l* fås nu normalvektoren $\vec{n}$ til planen α:

$\mathbf{n}:=\text{crossP}(\mathbf{v_l},\mathbf{pp_l}) ▸ \{4,6,-4\}$

Planens ligning findes

$\mathbf{eq\alpha}:=\text{dotP}(\{x,y,z\}-\mathbf{p_l},\mathbf{n})=0 ▸ 4\cdot x+6\cdot y-4\cdot z-32=0$

Planens ligning er altså givet ved

$\mathbf{eq\alpha} ▸ 4\cdot x+6\cdot y-4\cdot z-32=0$.

b) Koordinatsættet til skæringspunktet *Q* mellem linjen *m* og planen α findes ved at løse ligningssystemet bestående af linjens parameterfremstilling og planens

$x=\mathbf{m}[1] ▸ x=s+4$ $\quad y=\mathbf{m}[2] ▸ y=2\cdot s-4$ $\quad z=\mathbf{m}[3] ▸ z=2$

$$\mathbf{løsning}:=\text{solve}\left(\begin{cases}x=\mathbf{m}[1]\\ y=\mathbf{m}[2]\\ z=\mathbf{m}[3]\\ \mathbf{eq\alpha}\end{cases},x,y,z,s\right)$$

$▸ x=7 \text{ and } s=3 \text{ and } y=2 \text{ and } z=2$

$\mathbf{q}:=\mathbf{m}|\mathbf{løsning} ▸ \{7,2,2\}$

Skæringspunktet *Q* har koordinaterne

c) Vi finder gradtallet for vinklen *u* mellem retningsvektoren $\mathbf{v_m}$ for *m* og planens normalvektor $\vec{n}$ og trækker den fundne vinkel fra 90°:

$$\mathbf{u}:=90-\cos^{-1}\left(\frac{\text{dotP}(\mathbf{v_m},\mathbf{n})}{\text{norm}(\{\mathbf{v_m}\})\cdot\text{norm}(\{\mathbf{n}\})}\right) ▸ 60.195$$

Den spidse vinkel mellem *m* og α er altså

$\text{round}(\mathbf{u},1) ▸ 60.2$.

d) Den linje *k*, der går gennem *P* og skærer både *l* og *m*, må gå gennem *m*'s skæringspunkt *Q* med α. Vi skal altså finde en parameterfremstilling for linjen gennem *P* og *Q*:

$\mathbf{k}:=\mathbf{p}+t\cdot(\mathbf{q}-\mathbf{p}) ▸ \{2\cdot t+5,4-2\cdot t,3-t\}$

En mulig parameterfremstilling er altså givet ved

$[x;y;z]=(\{\mathbf{k}\})^{T} ▸ \begin{bmatrix}x=2\cdot t+5\\ y=4-2\cdot t\end{bmatrix}$

Bemærkning: Vi har i det ovenstående anvendt planens ligning på vektorform, dvs.

$$\text{dotP}(\{x,y,z\}-\mathbf{p_1},\mathbf{n})=0$$

i stedet for planens ligning på den sædvanlige symbolske form

$$a\cdot(x-x_0)+b\cdot(y-y_0)+c\cdot(z-z_0)=0 \ .$$

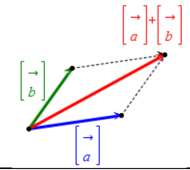
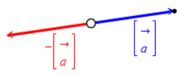
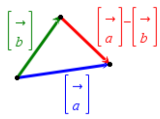
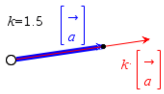
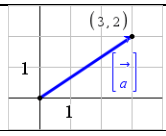
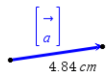
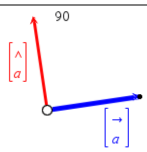
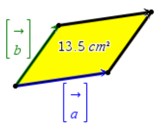
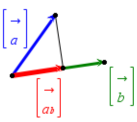
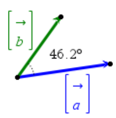
Det skyldes at det er ret bøvlet at substituere værdier for *a*, *b*, *c*, x_0, y_0 og z_0 i planens ligning på den symbolske form. Det bliver dermed også ret bøvlet at 'opgradere' besvarelsen til en fuldt interaktiv besvarelse, når man arbejder med planes ligning på den symbolske form.

Tegning af vektorer

Vi ser på 2-dimensional geometri, hvor vektorer kan tegnes med en geometri-kommando. Først en skematisk oversigt over hvordan man kan håndtere de mest almindelige regneoperationer geometrisk. De simple regneoperationer udføres typisk som transformationer, dvs. forskydninger, drejninger, multiplikationer/ligedannetheder osv. De er noteret med grønt. Men koordinater, længder, determinanter og vinkler findes ved målinger. De er noteret med blåt.

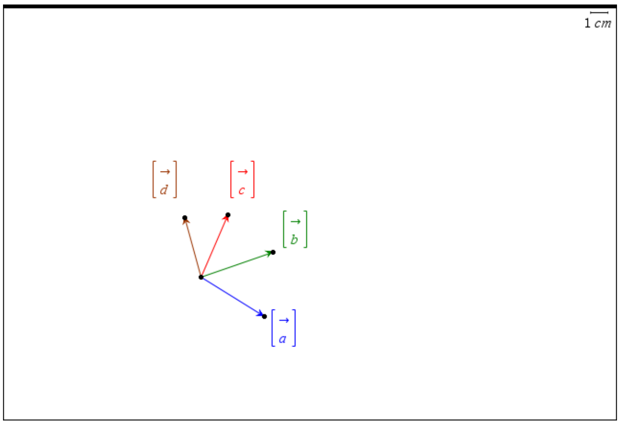
$\{\{2,1\}\}$ ▸Polar ▸

$\left[\sqrt{5}\ \ \angle\tan^{-1}\left(\frac{1}{2}\right)\right]$

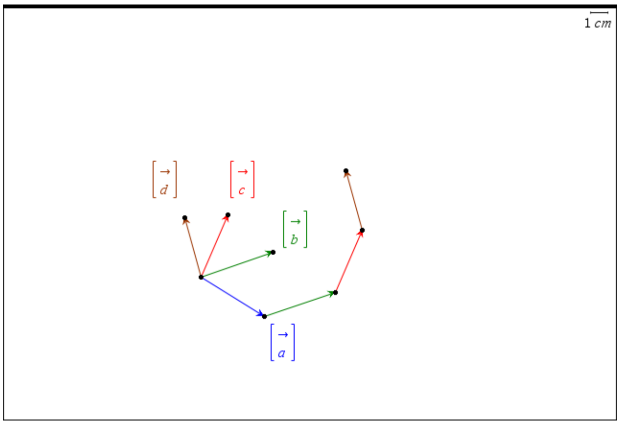
Operation	Fremgangsmåde	Billede	Kommentar
Summen af to vektorer $\vec{a}+\vec{b}$	Parallelforskydning af den ene vektor $\vec{b}$ langs den anden vektor $\vec{a}$		Kræfternes parallelogram! Kan derfor også konstrueres som en parallel konstruktionen med efterfølgende skæringspunkt.
Den **modsatte** vektor $-\vec{a}$	Spejling i et punkt		Kan også udføres som en multiplikation med skalafaktoren -1.
Differensen af to vektorer $\vec{a}-\vec{b}$	Parallelforskydning af modsat vektor $-\vec{b}$ langs vektor $\vec{a}$		Kan også udføres ved at forbinde $\vec{b}$'s endepunkt med $\vec{a}$'s endepunkt.
Produktet af vektor med en skalar $k\cdot\vec{a}$	Multiplikation af vektoren $\vec{a}$ med skalafaktoren k		Man kan også overføre produktet af k og længden af vektor $\vec{a}$ til vektor $\vec{a}$
Koordinaterne til en vektor $\vec{a}$	Koordinatværktøj ud fra punktkoordinater til stedvektor		Kan også konverteres til polære koordinater, dvs. længde og retningsvinkel.
Længden af en vektor $\|\vec{a}\|$	Måling af længden		Højreklik på vektoren og vælg **målinger**.
Tværvektoren til en vektor $\hat{a}$	Drejning af vektoren $\vec{a}$ med 90°		
Determinanten af to vektorer $\det(\vec{a},\vec{b})$	Måling af arealet af parallelogrammet udspændt af $\vec{a}$ og $\vec{b}$		Højreklik på parallelogram og vælg **målinger**. **Målingen giver kun den numeriske værdi!**
Projektion af en vektor $\vec{a}$ på en vektor $\vec{b}$	Vinkelret konstruktion med efterfølgende skæringspunkt		
Vinklen mellem to vektorer $\vec{a}$ og $\vec{b}$	Måling af vinklen		**Målingen giver kun den numeriske værdi!**

Lad os se på nogle eksempler.

Tip:
Tekstfeltet i **Geometri**-værkstedet understøtter ikke vektorpile. Men ved at pakke navnet ind i en søjlevektor kan man selv sætte en pil øverst og navnet nederst ved fx at skrive [→;a] eller benytte en skabelon.



Hvis vi fx har givet en stribe vektorer $\vec{a}, \vec{b}, \vec{c}$ og $\vec{d}$ afsat fra det samme punkt og vi skal konstruere summen $\vec{a}+\vec{b}+\vec{c}+\vec{d}$ kan vi gå således frem:



Vi skal have anbragt vektoren $\vec{b}$ i halen af vektoren $\vec{a}$. Vi vælger derfor parallelforskydning i **Transformations**-menuen. En parallelforskydning kan nu fastlægges på to forskellige måder:

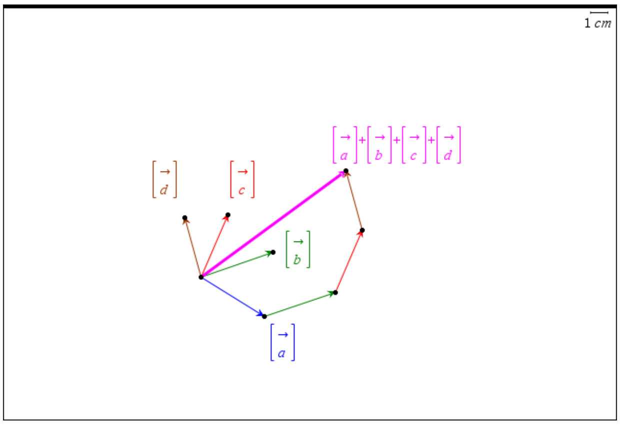
Via to forskydningspunkter: Vi klikker på start- og slutpunktet for forskydningen (og derefter på det objekt, der skal forskydes)

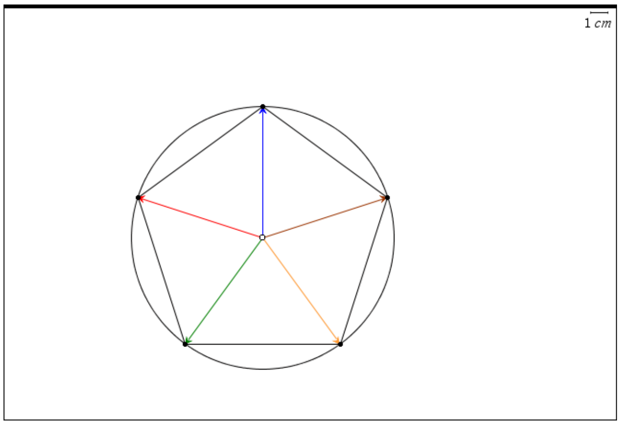
Via én forskydningsvektor: Vi klikker på forskydningsvektoren (og derefter på det objekt, der skal forskydes).

I dette tilfælde er det mest bekvemt at bruge to forskydningspunkter: For at afsætte vektor $\vec{b}$ i halen på vektor $\vec{a}$ vælger vi derfor parallelforskydning i **Transformations**-menuen og klikker derefter først på start- og slutpunkt for vektor $\vec{a}$ og derefter på vektor $\vec{b}$. Bagefter gentages forskydningen, men nu forskyder vi slutpunktet for vektor $\vec{b}$.

For derefter at hægte vektor $\vec{c}$ i enden på den forskudte vektor $\vec{b}$ vælger vi igen parallelforskydning fra **Transformations**-menuen, klikker først på startpunktet for vektor $\vec{a}$ og på slutpunktet for den forskudte vektor $\vec{b}$ og derefter på vektor $\vec{c}$ osv.

Til slut forbinder vi startpunktet for vektor $\vec{a}$ med slutpunktet for den sidst forskudte vektor $\vec{d}$ og vi har konstrueret summen af de fire vektorer, dvs. $\vec{a}+\vec{b}+\vec{c}+\vec{d}$.

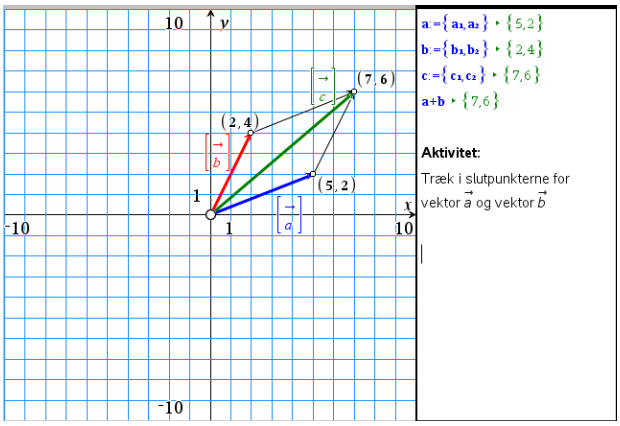




Det går ret hurtigt, når først man har fået det i fingrene. Så kan man fx prøve at konstruere summen af de fem radialvektorer i en regulær femkant.

På samme måde kan man lege med de andre regneoperationer. Faktisk kan man med fordel kombinere den grafiske fremstilling med en **Note**-side i en dynamisk interaktiv illustration. Det er da illustrativt, at begrænse koordinaterne til heltallige koordinater ved hjælp af et passende gitter. Her er det vist for summen af to vektorer. Her er vektoren $\vec{a}$'s koordinater lagret i variablene $\mathbf{a_1}$ og $\mathbf{a_2}$ og tilsvarende for de andre vektorers koordinater.

Tip
Man lagrer punkters koordinater ved at højreklikke på koordinaten og vælge **lagre** i kontekstmenuen.



Prøv nu selv at fremstille interaktive illustrationer af nogle af de andre regneoperationer for vektorer.

Til slut ser vi nu på en typisk opgave med 2-dimensionale vektorer, som vi nu kan løse både symbolsk og geometrisk:

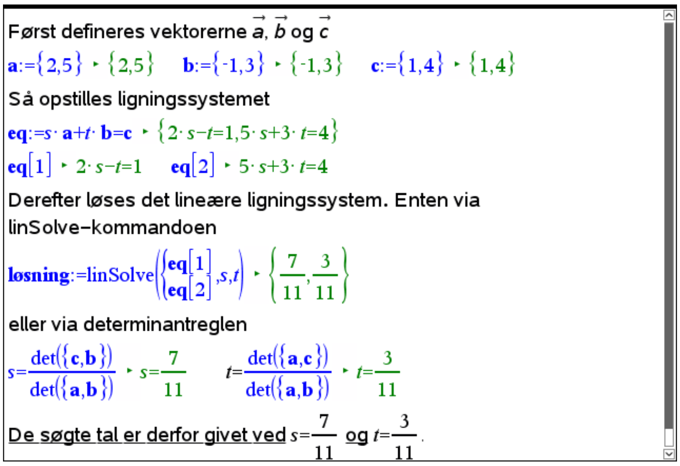
Der er givet vektorerne

$$\vec{a}=\begin{bmatrix}2\\5\end{bmatrix},\quad \vec{b}=\begin{bmatrix}-1\\3\end{bmatrix}\quad \text{og}\quad \vec{c}=\begin{bmatrix}1\\4\end{bmatrix}$$

Bestem tallene s og t så

$$s\cdot\vec{a}+t\cdot\vec{b}=\vec{c}$$

Først løser vi opgaven i **Note**-værkstedet. Hvis vi bruger determinantreglen skal vi lige huske at pakke liste-vektorerne ind i en krøllet parentes, fordi determinanten er en matrix-kommando


Først defineres vektorerne $\vec{a}$, $\vec{b}$ og $\vec{c}$

$\mathbf{a}:=\{2,5\} \blacktriangleright \{2,5\}$ $\quad \mathbf{b}:=\{-1,3\} \blacktriangleright \{-1,3\}$ $\quad \mathbf{c}:=\{1,4\} \blacktriangleright \{1,4\}$

Så opstilles ligningssystemet

$\mathbf{eq}:=s\cdot \mathbf{a}+t\cdot \mathbf{b}=\mathbf{c} \blacktriangleright \{2\cdot s-t=1, 5\cdot s+3\cdot t=4\}$

$\mathbf{eq}[1] \blacktriangleright 2\cdot s-t=1 \quad \mathbf{eq}[2] \blacktriangleright 5\cdot s+3\cdot t=4$

Derefter løses det lineære ligningssystem. Enten via linSolve-kommandoen

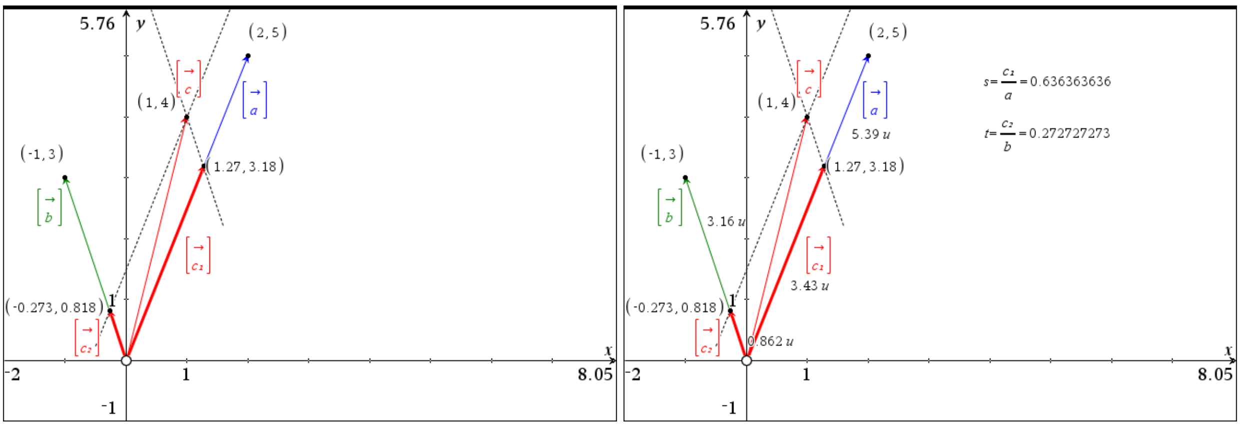
$\mathbf{løsning}:=\text{linSolve}\left(\begin{cases}\mathbf{eq}[1]\\ \mathbf{eq}[2]\end{cases}, s, t\right) \blacktriangleright \left\{\frac{7}{11}, \frac{3}{11}\right\}$

eller via determinantreglen

$s=\frac{\det(\{\mathbf{c},\mathbf{b}\})}{\det(\{\mathbf{a},\mathbf{b}\})} \blacktriangleright s=\frac{7}{11} \qquad t=\frac{\det(\{\mathbf{a},\mathbf{c}\})}{\det(\{\mathbf{a},\mathbf{b}\})} \blacktriangleright t=\frac{3}{11}$

De søgte tal er derfor givet ved $s=\frac{7}{11}$ og $t=\frac{3}{11}$.

Så løser vi den geometrisk ved at afsætte vektorerne $\vec{a}, \vec{b}$ og $\vec{c}$ i et koordinatsystem. Vi skal da opløse vektoren $\vec{c}$ i to komponenter, der er parallelle med henholdsvis vektor $\vec{a}$ og vektor $\vec{b}$. Vi trækker derfor paralleller og opløser i første omgang vektor $\vec{c}$ som en sum af to vektorer $\vec{c_1}$ og $\vec{c_2}$, hvor $\vec{c_1}$ er parallel med vektor $\vec{a}$ og $\vec{c_2}$ er parallel med vektor $\vec{b}$.



Derefter finder vi skalarerne *s* og *t* som forholdet mellem længderne af vektorerne, dvs.

$$s=\frac{|\vec{c_1}|}{|\vec{a}|} \quad \text{og} \quad t=\frac{|\vec{c_2}|}{|\vec{b}|}$$

Vi måler altså længderne af de involverede vektorer og indskriver tekstboksene

$$\boxed{\frac{c_1}{a}} \quad \text{og} \quad \boxed{\frac{c_2}{b}}$$

Derefter udfører vi en beregning af tekstboksene ved at udpege de relevante længder.

Obs
Det er brugerens ansvar selv at holde styr på fortegnene for skalafaktorerne *s* og *t*. Hvis vektorerne er ensrettede er skalafaktoren positiv. Er de derimod modsatrettede er skalafaktoren negativ.

Denne gang finder vi kun tilnærmede værdier, men i et **Note**-værksted kan de konverteres til eksakte værdier, da der tydeligvis er tale om periodiske decimalbrøker med perioden 2.

$$\text{exact}(0.636363636,2) \blacktriangleright \frac{7}{11}$$

$$\text{exact}(0.272727273,2) \blacktriangleright \frac{3}{11}$$

Endelig bemærker vi, at den ovenstående opgave sagtens kan løses symbolsk i 3 dimensioner efter nøjagtigt de samme principper. Prøv fx selv kræfter med

Der er givet vektorerne

$$\vec{a} = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix}, \quad \vec{c} = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} \quad \text{og} \quad \vec{d} = \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix}$$

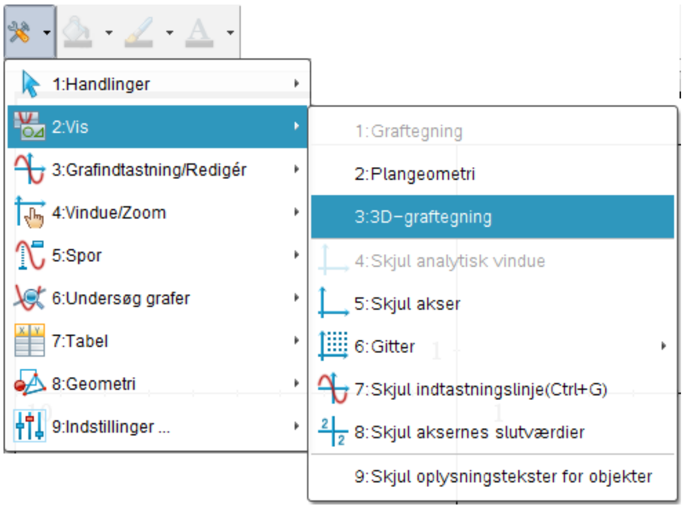
Bestem tallene *r, s* og *t* så

$$r \cdot \vec{a} + s \cdot \vec{b} + t \cdot \vec{c} = \vec{d}$$

Derimod kan vi *ikke* løse den geometrisk i TI-Nspire CAS fordi TI-Nspire CAS ikke for tiden understøtter 3-dimensional geometri. TI-Nspire CAS understøtter dog 3-dimensionale grafer, så vi kan godt illustrere opgaver i 3-dimensionale vektorer med passende grafer. Det vender vi os nu mod ☺

*Rumgeometri**

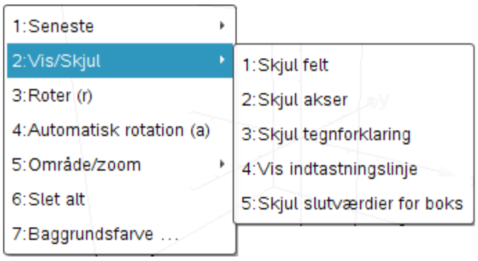
Vi skal nu se nærmere på hvad vi kan illustrere i **3d-graf**-værkstedet. Vi åbner derfor et **Graf**-værksted og skifter til **3D-graftegning** i **Vis**-menuen:



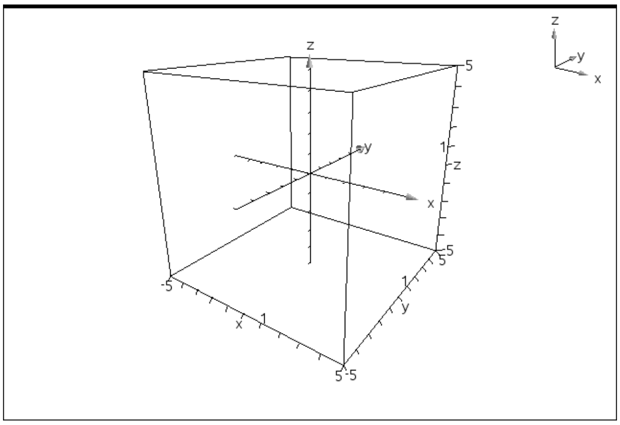
Der åbnes da for en 3-dimensional scene inde i en terningeformet boks. Kun den del af tegneobjekterne, der ligger indenfor scenen vil blive vist. Resten vil blive klippet.

Scenen er tegnet i centralt perspektiv, dvs. bagenden af scenerummet fremstår mindre end forenden af scenerummet.

Man kan slukke for scenen, ligesom man kan slukke for koordinatsystemet og for signaturen, dvs. det lille ikoniske koordinatsystem, der viser orienteringen af det store koordiantsystem. Fx kan man højreklikke i scenen og gå ind og vælge vis/skjul-menuen:



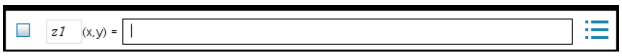
Endelig skal man lægge mærke til at man kan dreje scenen til siden eller op og ned ved at trække med musen. Det 3-dimensionale scenerum er altså fuldt interaktivt.

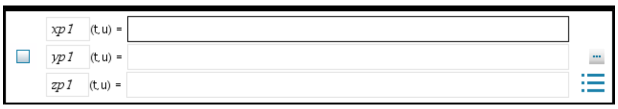


Det er på denne scene, vi nu skal tegne punkter, linjer planer osv.

Punkter og kugler

Der er to forskellige graftyper: Funktioner af 2 variable og parameterfremstillinger.





Vi skal især bruge parameterfremstillinger. I princippet kunne vi nu tegne et punkt ved simpelthen at indtaste koordinaterne i xp1, yp1 og zp1. Men man kan faktisk ikke se punktet ☺. Det er for tyndt og skal først fedes op. Vi tegner derfor i stedet for et punkt som en lille kugle. Vi starter med at se på hvordan man tegner en enhedskugle i rummet. Vi splitter da koordinatsættet i en vandret del og en lodret del

$$\{x, y, z\} = \{x, y, 0\} + \{0, 0, z\}$$

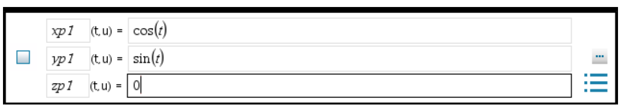
Ækvatorcirklen i x-y-planen er da givet ved

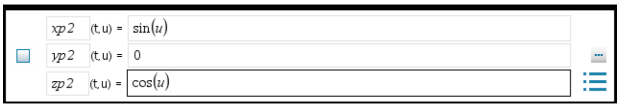
$$\{\cos(t), \sin(t), 0\} \quad 0 \le t \le 2\pi$$

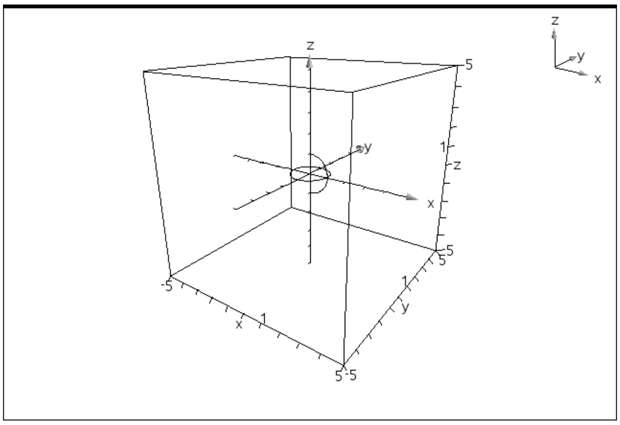
Tilsvarende er *hovedmeridianen* i z-x-planen givet ved

$$\{\sin(u), 0, \cos(u)\} \quad 0 \le u \le \pi$$

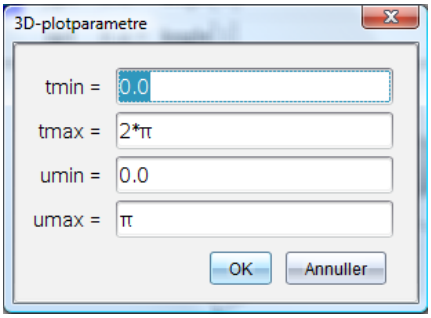
Sådanne parameterfremstillinger kan indskrives direkte i indtastningslinjen i 3d-grafrummet, så vi kan få tegnet ækvatorcirklen og hovedmeridianen







Man kan måske undre sig over hvordan TI-Nspire CAS kan vide at ækvatorcirklen skal tegnes som en hel cirkel, mens hovedmeridianen skal tegnes som en halvcirkel. Men klikker vi på parameter-ikonet ··· ser du netop disse valg:



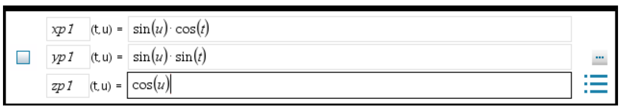
Som standard er parameteren *t* altså valgt til at fungere som en ækvatorvinkel, mens parameteren *u* er valgt til at fungere som en polvinkel. Men for andre geometriske objekter må vi selvfølgelig gå ind og rette parameterintervallerne til, hvad der nu måtte passe under disse omtændigheder.

Vi kombinerer nu de to parameterfremstillinger for ækvatorcirklen og hovedmeridianen til en enkelt samlet parameterfremstilling for enhedskuglen

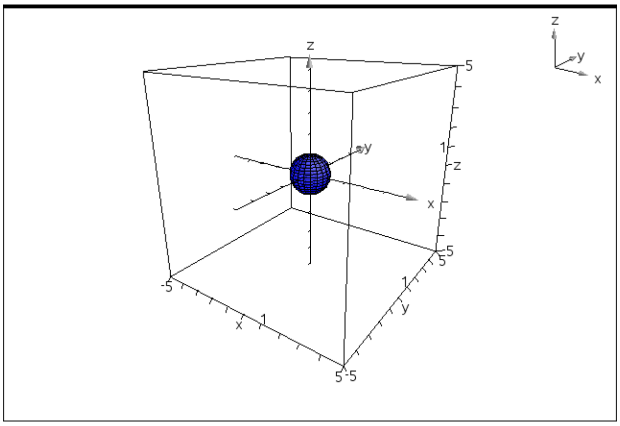
$$\sin(u)\cdot\{\cos(t),\sin(t),0\}+\cos(u)\cdot\{0,0,1\}=\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\}$$

$$0\leq t\leq 2\pi,\quad 0\leq u\leq \pi$$

Vi kan derfor tegne en enhedskugle ved at indskrive denne parameterfremstilling direkte i indtastningslinjen

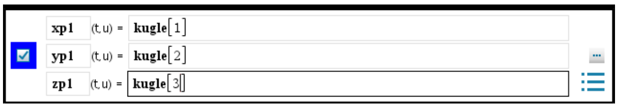


Det fører straks til tegning af en enhedskugle



Det er imidlertid ikke en hensigtsmæssig måde at håndtere geometriske objekter på, da alle rettelser så skal foregå inde i indtastningslinjen for **3d-graf**-rummet. I stedet opretter vi en definition på en **Note**-side og referer til den på indtastningslinjen:

$$\textbf{kugle}:=\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\} \blacktriangleright \{\cos(t)\cdot\sin(u),\sin(t)\cdot\sin(u),\cos(u)\}$$



Vi er nu meget tæt på at kunne tegne punkter. Det kan da betale sig at indføre en generel punkt-kommando:

$$\textbf{kugle}:=\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\} \blacktriangleright \{\cos(t)\cdot\sin(u),\sin(t)\cdot\sin(u),\cos(u)\}$$

$$\textbf{punkt}(pkt,radius):=pkt+radius\cdot\textbf{kugle} \blacktriangleright \textit{Udført}$$

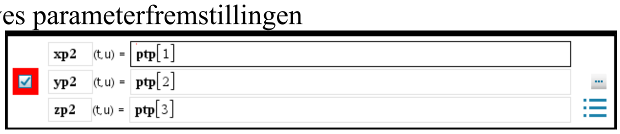
Det anbefales at starte en hver ny opgave i rumgeometri med disse to linjer!

Hvis vi fx vil have tegnet punktet $P=(2,3,4)$ skriver vi da blot

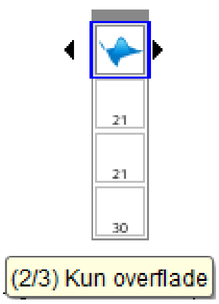
$$\textbf{p}:=\{2,3,4\} \blacktriangleright \{2,3,4\}$$

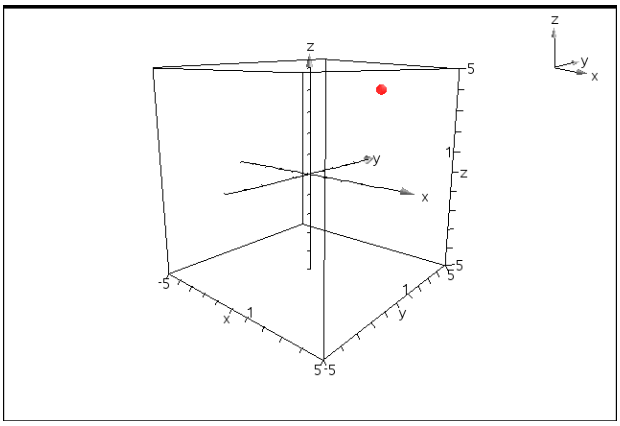
$$\textbf{ptp}:=\textbf{punkt}(\textbf{p},0.25) \blacktriangleright \{0.25\cdot\cos(t)\cdot\sin(u)+2,0.25\cdot\sin(t)\cdot\sin(u)+3,0.25\cdot\cos(u)+4\}$$

Derefter indskrives parameterfremstillingen



Læg mærke til at punktet kan farvelægges ved at klikke på det, og at gitteret kan slås fra ved at højreklikke og vælge attributter. Så fremstår farven klarere.

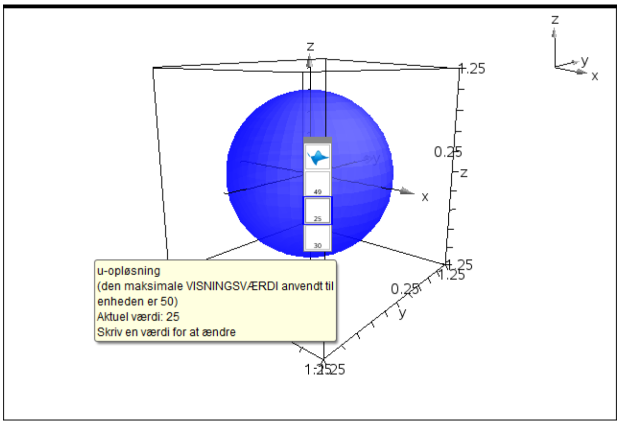




Så nu kan vi tegne såvel punkter som kugler ☺. Vi kan da fx bruge det til at lege med kuglens parameterfremstilling. Vi opretter da et 3d-grafrum med en enhedskugle men skifter områdeindstillinger til

$$-1.25 \le x \le 1.25, \quad -1.25 \le y \le 1.25, \quad -1.25 \le z \le 1.25$$

Obs
I **3d-Graf**-værkstedet hedder det ikke vin-duesindstillinger, men områdeindstil-linger, men de findes på sædvanlig vis fx ved at højreklikke i grafvinduet.

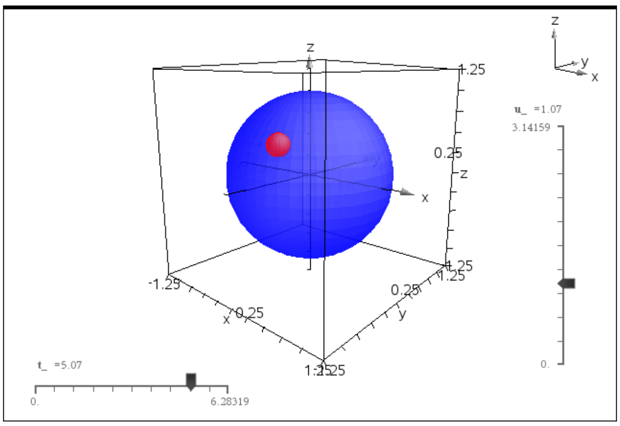


Vi indfører nu to skydere for henholdsvis længdegraden t_ og breddegraden u_ med parameterintervallerne $0 \le t\_ \le 2\pi$ henholdsvis $0 \le u\_ \le \pi$ og endelig afbilder vi punktet

$$p:=\{\sin(u\_)\cdot\cos(t\_),\sin(u\_)\cdot\sin(t\_),\cos(u\_)\} \rightarrow \{\sin(1)\cdot\cos(1),(\sin(1))^2,\cos(1)\}$$

$$ptp:=punkt(p,0.125)$$

$$\rightarrow \{0.125\cdot\cos(t)\cdot\sin(u)+\sin(1)\cdot\cos(1),0.125\cdot\sin(t)\cdot\sin(u)+(\sin(1))^2,0.125\cdot\cos(u)+\cos(1)\}$$



Når vi trækker i $t\_$-skyderen bevæger punktet sig da vandret rundt langs en breddegrad, og når vi tilsvarende trækker i $u\_$-skyderen bevæger punkter sig lodret op og ned langs en længdegrad.

Linjer, linjestykker og halvlinjer

En linje gennem to punkter A og B er givet ved parameterfremstillingen

$$A + t \cdot \overrightarrow{AB} = A + t \cdot (B - A) = (1-t) \cdot A + t \cdot B, \quad -\infty < t < \infty$$

Læg mærke til, at summen af koefficienterne til A og B netop er 1. Det kaldes en *konveks kombination*. Det har den fordel, at den er uafhængig af valget af Origo, dvs. konvekse kombinationer giver mening i en Euklidisk plan. Sætter vi fx $t = \frac{1}{2}$ fås fx midtpunktet på linjestykket AB, mens begyndelsespunktet A svarer til $t = 0$ og enhedspunktet B svarer til $t = 1$.

Parameterfremstillingen kan skrives direkte ind, bortset fra at parameterintervallet selvfølgelig ikke kan række fra $-\infty$ til ∞. I praksis er det typisk nok, at sætte parameterintervallet til $-10 \le t \le 10$.

Tilsvarende er linjestykket fra A til B givet ved

$$(1-t) \cdot A + t \cdot B, \quad 0 \le t \le 1$$

Og endelig er halvlinjen med toppunkt i A som går gennem B givet ved

$$(1-t) \cdot A + t \cdot B, \quad 0 \le t < \infty$$

Skal vi fx tegne linjen gennem punkterne $A = (1,2,3)$ og $B = (2,-1,-2)$ ser det derfor således ud (husk at rette parameterintervallet til at gå fra –10 til 10!)

kugle$:=\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\}$ ▸ $\{\cos(t)\cdot\sin(u),\sin(t)\cdot\sin(u),\cos(u)\}$

punkt$(pkt,radius):=pkt+radius\cdot$ **kugle** ▸ *Udført*

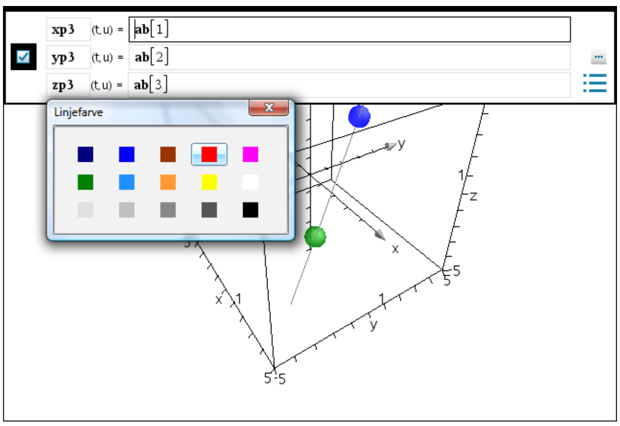
a$:=\{1,2,3\}$ ▸ $\{1,2,3\}$

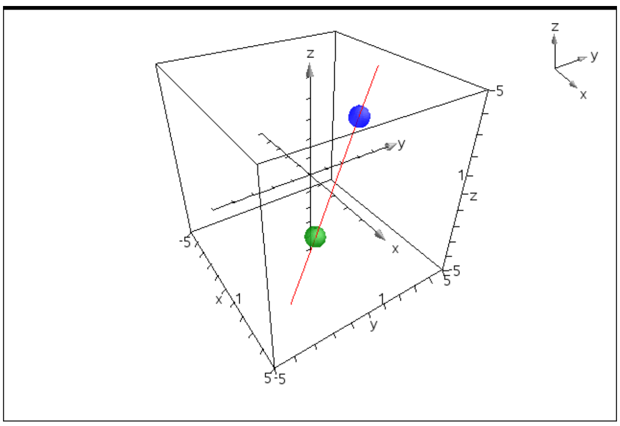
b$:=\{2,-1,-2\}$ ▸ $\{2,-1,-2\}$

pta$:=$**punkt**$\left(\mathbf{a},\frac{1}{2}\right)$ ▸ $\left\{\frac{\cos(t)\cdot\sin(u)}{2}+1,\frac{\sin(t)\cdot\sin(u)}{2}+2,\frac{\cos(u)}{2}+3\right\}$

ptb$:=$**punkt**$\left(\mathbf{b},\frac{1}{2}\right)$ ▸ $\left\{\frac{\cos(t)\cdot\sin(u)}{2}+2,\frac{\sin(t)\cdot\sin(u)}{2}-1,\frac{\cos(u)}{2}-2\right\}$

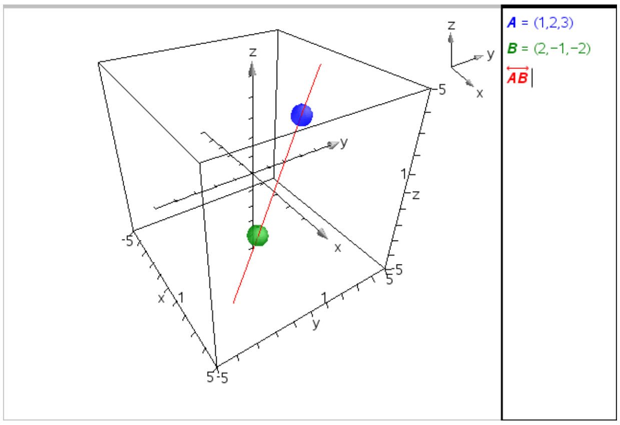
ab$:=(1-t)\cdot\mathbf{a}+t\cdot\mathbf{b}$ ▸ $\{t+1,2-3\cdot t,3-5\cdot t\}$





Læg mærke til at vi ikke kan fange linjen ved at klikke på den. Skal vi skifte farve på linjen skal vi derfor fange den i indtastningslinjen, hvor vi kan højreklikke og vælge **farve**-menuen som vist.

Læg også mærke til, at vi ikke kan navngive objekter inde i **3d-Graf**-rummet. I stedet kan vi tilføje en smal **Note**-bjælke og vise farvekodernes betydning der ☺



Vektorer

Det er desværre ikke helt nemt at tegne vektorer korrekt. Det er pilespidsen der volder problemer, da den skal tegnes som en lille kegle. I stedet kan man tegne en fattigmands-vektor som en *knappenål*, dvs. som et begyndelsespunkt og et linjestykke. Nålens hoved er da vektorens begyndelsespunkt, som man kan gribe fat i, mens man kan stikke med linjestykket.

En vektor $\overrightarrow{AB}$ tegnes altså som punktet *A* efterfulgt af linjestykket *AB*. Det er selvsagt ikke lige så flot som en rigtig vektor, men kan godt give en god fornemmelse for hvordan vektoren ligger i rummet.

Planer, parallelogrammer og halvplaner

En plan gennem tre punkter A, B og C er givet ved parameterfremstillingen

$$A + t\cdot(B-A) + u\cdot(C-A) = (1-t-u)\cdot A + t\cdot B + u\cdot C,$$
$$-\infty < t < \infty, \quad -\infty < u < \infty$$

Læg mærke til, at summen af koefficienterne til *A*, *B* og *C* netop er 1. Det kaldes en *konveks kombination*. Det har den fordel, at den er uafhængig af valget af Origo, dvs. konvekse kombinationer giver mening i en Euklidisk plan. Sætter vi fx $t = \frac{1}{3}, u = \frac{1}{3}$ fås tyngdepunktet for trekanten *ABC*, mens begyndelsespunktet *A* svarer til $t = 0, u = 0$ og enhedspunkterne *B* og *C* svarer til $t = 1, u = 0$ henholdsvis $t = 0, u = 1$.

Parameterfremstillingen kan skrives direkte ind, bortset fra at parameterintervallet selvfølgelig ikke kan række fra $-\infty$ til ∞. I praksis er det typisk nok at sætte parameterintervallet til at gå fra –10 til 10.

Tilsvarende er parallelogrammet udspændt af linjestykkerne AB og AC givet ved

$$(1-t-u)\cdot A + t\cdot B + u\cdot C, \quad 0 \le t \le 1, \quad 0 \le u \le 1$$

Og endelig er halvplanen med kant langs AB som går gennem C givet ved

$$(1-t-u)\cdot A + t\cdot B + u\cdot C, \quad -\infty < t < \infty, \quad 0 \le u < \infty$$

Skal vi fx tegne planen gennem punkterne $A=(1,2,3)$, $B=(2,-1,-2)$ og $C=(-2,3,1)$ ser det derfor således ud

kugle:= $\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\}$ ▸ $\{\cos(t)\cdot\sin(u),\sin(t)\cdot\sin(u),\cos(u)\}$

punkt(*pkt*,*radius*):=*pkt*+*radius*· **kugle** ▸ *Udført*

a:= $\{1,2,3\}$ ▸ $\{1,2,3\}$

b:= $\{2,-1,-2\}$ ▸ $\{2,-1,-2\}$

c:= $\{-2,3,1\}$ ▸ $\{-2,3,1\}$

pta:=**punkt**$\left(\mathbf{a},\frac{1}{2}\right)$ ▸ $\left\{\frac{\cos(t)\cdot\sin(u)}{2}+1,\frac{\sin(t)\cdot\sin(u)}{2}+2,\frac{\cos(u)}{2}+3\right\}$

ptb:=**punkt**$\left(\mathbf{b},\frac{1}{2}\right)$ ▸ $\left\{\frac{\cos(t)\cdot\sin(u)}{2}+2,\frac{\sin(t)\cdot\sin(u)}{2}-1,\frac{\cos(u)}{2}-2\right\}$

ptc:=**punkt**$\left(\mathbf{c},\frac{1}{2}\right)$ ▸ $\left\{\frac{\cos(t)\cdot\sin(u)}{2}-2,\frac{\sin(t)\cdot\sin(u)}{2}+3,\frac{\cos(u)}{2}+1\right\}$

abc:= $(1-t-u)\cdot\mathbf{a}+t\cdot\mathbf{b}+u\cdot\mathbf{c}$ ▸ $\{t-3\cdot u+1,-3\cdot t+u+2,-5\cdot t-2\cdot u+3\}$

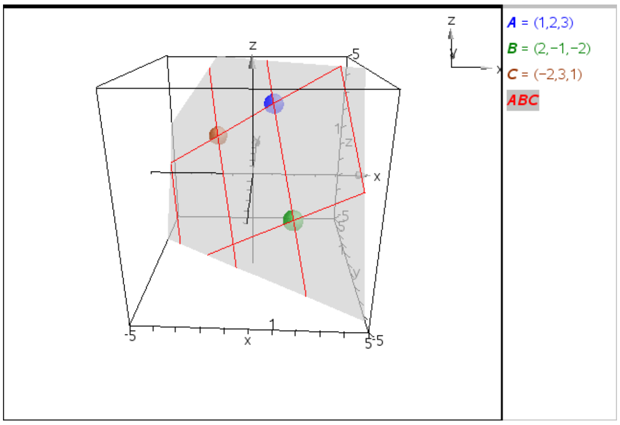


Illustration af en vektoropgave

Nu, da vi har de mest fundamentale geometriske objekter på plads, kan vi prøve at illustrere den første af de to vektoropgaver, som vi tidligere har løst analytisk.

I et koordinatsystem i rummet er der givet 3 vektorer

$$\vec{a}=\begin{pmatrix}1\\2\\3\end{pmatrix},\quad \vec{b}=\begin{pmatrix}2\\-1\\2\end{pmatrix},\quad \vec{c}=\begin{pmatrix}-1\\2\\3\end{pmatrix}$$

a) Bestem et gradtal for vinklen mellem $\vec{a}$ og $\vec{b}$.

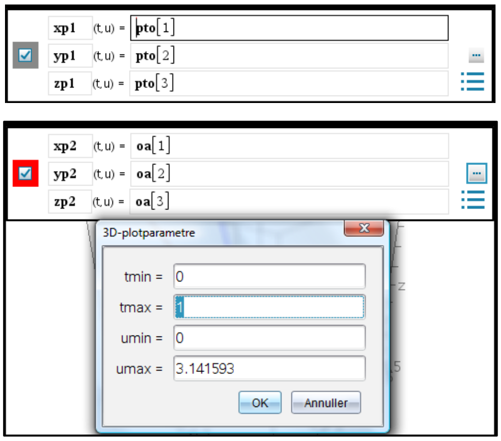
b) Bestem koordinatsættet til projektionen af $\vec{a}$ på $\vec{c}$.

c) Bestem tallene *s* og *t*, således at vektoren

$$\vec{d}=\vec{a}+s\cdot\vec{b}+t\cdot\vec{c}$$

står vinkelret på både $\vec{b}$ og $\vec{c}$, og angiv koordinaterne for $\vec{d}$.

Vi starter da med at tegne punktet O og vektorerne $\vec{a}$, $\vec{b}$ og $\vec{c}$ (husk at rette parameter-intervallerne til for linjestykkerne OA, OB og OC):



$\textbf{kugle}:=\{\sin(u)\cdot\cos(t),\sin(u)\cdot\sin(t),\cos(u)\} \blacktriangleright \{\cos(t)\cdot\sin(u),\sin(t)\cdot\sin(u),\cos(u)\}$

$\textbf{punkt}(pkt,radius):=pkt+radius\cdot\textbf{kugle} \blacktriangleright$ *Udført*

$\textbf{o}:=\{0,0,0\} \blacktriangleright \{0,0,0\}$ $\textbf{a}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$ $\textbf{b}:=\{2,-1,2\} \blacktriangleright \{2,-1,2\}$

$\textbf{c}:=\{-1,2,3\} \blacktriangleright \{-1,2,3\}$

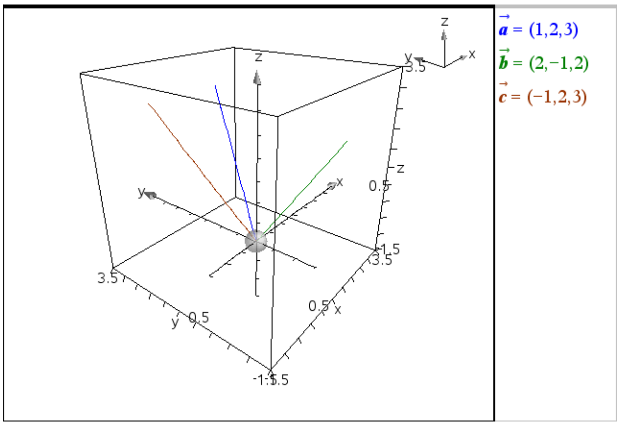
$\textbf{d}:=\left\{\frac{49}{61},\frac{56}{61},\frac{-21}{61}\right\} \blacktriangleright \left\{\frac{49}{61},\frac{56}{61},\frac{-21}{61}\right\}$ $\textbf{a}_c:=\left\{\frac{-6}{7},\frac{12}{7},\frac{18}{7}\right\} \blacktriangleright \left\{\frac{-6}{7},\frac{12}{7},\frac{18}{7}\right\}$

$\textbf{pto}:=\textbf{punkt}\left(\textbf{o},\frac{1}{4}\right) \blacktriangleright \left\{\frac{\cos(t)\cdot\sin(u)}{4},\frac{\sin(t)\cdot\sin(u)}{4},\frac{\cos(u)}{4}\right\}$

$\textbf{oa}:=(1-t)\cdot\textbf{o}+t\cdot\textbf{a} \blacktriangleright \{t,2\cdot t,3\cdot t\}$ $\textbf{ob}:=(1-t)\cdot\textbf{o}+t\cdot\textbf{b} \blacktriangleright \{2\cdot t,-t,2\cdot t\}$

$\textbf{oc}:=(1-t)\cdot\textbf{o}+t\cdot\textbf{c} \blacktriangleright \{-t,2\cdot t,3\cdot t\}$ $\textbf{aa}_c:=(1-t)\cdot\textbf{a}+t\cdot\textbf{a}_c \blacktriangleright \left\{1-\frac{13\cdot t}{7},2-\frac{2\cdot t}{7},3-\frac{3\cdot t}{7}\right\}$

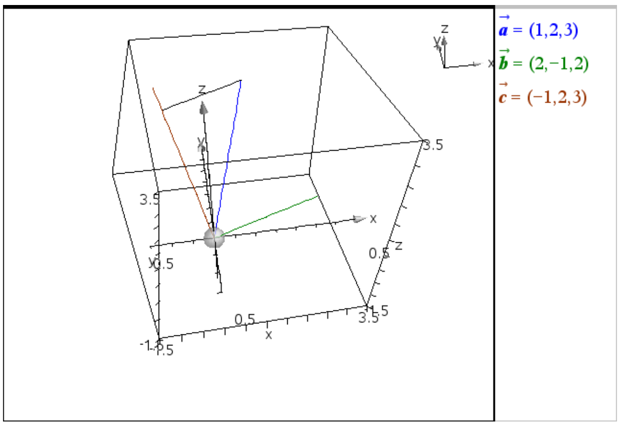
Bemærkning: Vi trækker på den analytiske løsnng, når vi fx indskriver koordinaterne til vektoren $\vec{d}$.



For at kunne vurdere størrelsen af vinklen mellem vektorerne $\vec{a}$ og $\vec{b}$ har vi drejet scenen, så vi så vidt muligt kigger vinkelret ind på vektorerne. Det ses at vinklen er klart større end 45°, men en præcis opmåling er ikke mulig.

Vi kan illustrere projektionen af vektor $\vec{a}$ på vektor $\vec{c}$ ved at tilføje den vinkelrette fra $\vec{a}$'s endepunkt på vektor $\vec{c}$.

$$\textbf{aa}_c:=(1-t)\cdot\textbf{a}+t\cdot\textbf{a}_c \blacktriangleright \left\{1-\frac{13\cdot t}{7},2-\frac{2\cdot t}{7},3-\frac{3\cdot t}{7}\right\}$$



Igen har vi drejet scenen, så den rette vinkel fremstår tydeligere.

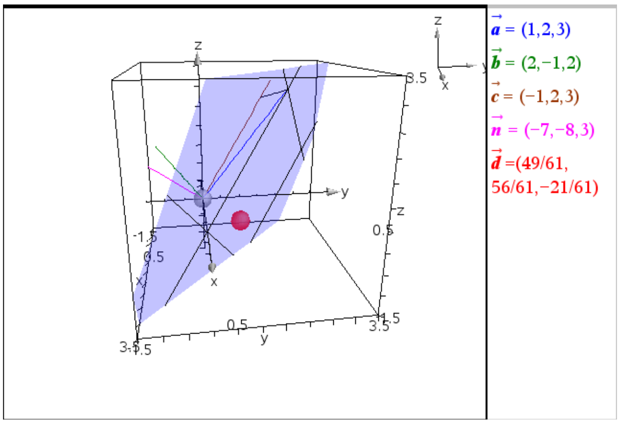
Endelig skal vi have illustreret vektoren $\vec{d}$, der er valgt så den står vinkelret på både $\vec{b}$ og $\vec{c}$. Vi kan da med fordel tilføje krydsproduktet $\vec{n} = \vec{b} \times \vec{c}$. Krydsproduktet bliver for langt til at det kan vises indenfor scenen, men det er jo kun retningen vi for alvor er interesserede i, så det gør ikke så meget ☺. Vi ved da at vektoren $\vec{d}$ skal pege i retning af vektoren $\vec{n}$.

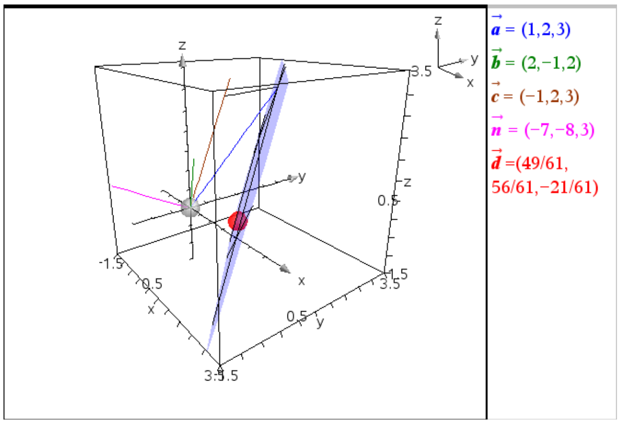
Skifter vi nu parameteren *s* ud med parameteren *u*, så vil *D* netop ligge i planen udspændt af *A*, *E* = *A*+*B* og *F* = *A*+*C*. Vi tegner derfor dels punktet *D* (dvs. slutpunktet for vektoren $\vec{d}$), dels planen *AEF*.

$$\mathbf{n}:=\text{crossP}(\mathbf{b},\mathbf{c}) \blacktriangleright \{-7,-8,3\} \qquad \mathbf{on}:=(1-t)\cdot \mathbf{o}+t\cdot \mathbf{n} \blacktriangleright \{-7\cdot t,-8\cdot t,3\cdot t\}$$

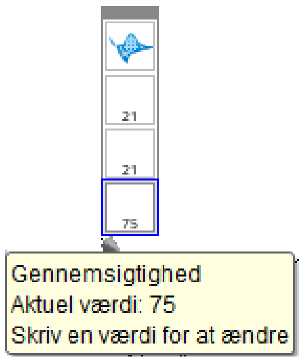
$$\mathbf{ptd}:=\mathbf{punkt}\left(\mathbf{d},\frac{1}{4}\right) \blacktriangleright \left\{\frac{\cos(t)\cdot \sin(u)}{4}+\frac{49}{61},\frac{\sin(t)\cdot \sin(u)}{4}+\frac{56}{61},\frac{\cos(u)}{4}-\frac{21}{61}\right\}$$

$$\mathbf{aef}:=\mathbf{a}+u\cdot \mathbf{b}+t\cdot \mathbf{c} \blacktriangleright \{-t+2\cdot u+1,2\cdot t-u+2,3\cdot t+2\cdot u+3\}$$





Man er da nødt til at dreje scenen passende for at kunne se at punktet *D* faktisk ligger i planen, ligesom det ligger i forlængelse af krydsproduktet $\vec{n} = \vec{b} \times \vec{c}$. Det kan også betale sig at gøre planen mere gennemsigtig/transparent. Det er den tredje attribut, vi i så fald skal skrue op for. Her er transparensen sat til 75.



Appendix: Vektorer som datastrukturer*

Koordinater til vektorer

Hvis man skal trække en koordinat ud fra en liste-vektor sker det helt simpelt. Fx

$$\mathbf{u}[2] \blacktriangleright 2$$

Her læses symbolet **u**[2] som andenkoordinaten til **u**. Læg mærke til, at man trækker koordinater ud ved hjælp af kantede parenteser.

Hvis man tilsvarende vil trække koordinater ud af række-vektorer eller søjle-vektorer bliver det mere indviklet. Vi minder om at en tabel (matrix) ikke er andet end en liste af lister. En liste af data er en *en-dimensional datastruktur* (ligesom en tallinje), mens en liste af lister af data udgør en *to-dimensional datastruktur* (ligesom et koordinatsystem). TI-Nspire CAS omdanner automatisk listen af lister til en tabel, fx

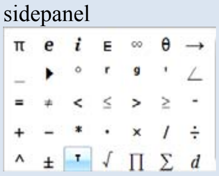
$$t:=\{\{1,2\},\{3,4\},\{5,6\}\}$$

der omformes til

$$\mathrm{t}:=\begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix} \blacktriangleright \begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix}$$

hvor de tre lister inde i listen skrives op under hinanden.

Tip
Transponeringstegnet $^{\mathsf{T}}$ findes i **tegn-oversigten** i venstre sidepanel



På samme måde er række-vektorer og søjle-vektorer reelt lister af lister, dvs. 2-dimensionale datastrukturer. Hvis vi starter med en liste-vektor kan den derfor omdannes til en række-vektor ved at sætte en krøllet parentes uden om, fx

$$\mathbf{u}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$$

$$\mathbf{v}:=\{\mathbf{u}\}$$

der omformes til

$$\mathbf{u}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$$

$$\mathbf{v}:=\{\mathbf{u}\} \blacktriangleright \begin{bmatrix}1&2&3\end{bmatrix}$$

Hvis vi starter med en liste-vektor kan den tilsvarende omdannes til en søjlevektor ved at sætte en krøllet parentes inden i, hvilket er lidt mere kompliceret ☺

$$\mathbf{w}:=\{\{1\},\{2\},\{3\}\}$$

Derved omformes udtrykket til

$$\mathbf{w}:=\{\{1\},\{2\},\{3\}\} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix}$$

I praksis bruger man derfor transponeringskommandoen

$$\mathbf{w}:=(\{\mathbf{u}\})^{\mathsf{T}} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix}$$

Hvad er så pointen? Jo række-vektorer og søjle-vektorer er i virkeligheden 2-dimensionale datastrukturer og det gør det kompliceret at trække koordinaterne ud. Hvis vi fx skal trække en koordinat ud af en række-vektor skal vi huske på, at en række-vektor i virkeligheden er en tabel/matrix med én række og tre søjler

$$\begin{array}{rc} & \begin{array}{ccc}\text{Søjle 1} & \text{Søjle 2} & \text{Søjle 3} \\ \downarrow & \downarrow & \downarrow\end{array} \\ \text{Række 1} \rightarrow & \begin{bmatrix}1 & 2 & 3\end{bmatrix}\end{array}$$

Ønsker vi fx at trække anden-koordinaten ud af en række-vektor ser det derfor således ud

$$\mathbf{v} := \begin{bmatrix}1 & 2 & 3\end{bmatrix} \blacktriangleright \begin{bmatrix}1 & 2 & 3\end{bmatrix}$$

$$\mathbf{v}[1,2] \blacktriangleright 2$$

Læg mærke til, at det første tal angiver række-nummeret, det andet søjle-nummeret (række kommer før søjle, ligesom bogstavet r kommer før bogstavet s).

Hvis vi tilsvarende skal trække en koordinat ud af en søjle-vektor skal vi huske på at en søjle-vektor i virkeligheden er en tabel/matrix med tre rækker og én søjle

$$\begin{array}{rc} & \begin{array}{c}\text{Søjle 1} \\ \downarrow\end{array} \\ \begin{array}{r}\text{Række 1} \rightarrow \\ \text{Række 2} \rightarrow \\ \text{Række 3} \rightarrow\end{array} & \begin{bmatrix}1 \\ 2 \\ 3\end{bmatrix}\end{array}$$

Ønsker vi fx at trække anden-koordinaten ud af en søjle-vektor ser det derfor således ud

$$\mathbf{w} := \begin{bmatrix}1 \\ 2 \\ 3\end{bmatrix} \blacktriangleright \begin{bmatrix}1 \\ 2 \\ 3\end{bmatrix}$$

$$\mathbf{w}[2,1] \blacktriangleright 2$$

Læg igen mærke til, at det første tal angiver række-nummeret, det andet søjle-nummeret.

Håndtering af matrix-kommandoer for liste-vektorer

Er der slet ingen ulemper med liste-vektorer? Jo, men der er ikke mange, så længe man bliver indenfor geometrien. Den vigtigste mangel, er at der *ikke* findes en liste-kommando til udregning af længden af en vektor.

Indenfor den lineære algebra kan man imidlertid bruge norm-kommandoen, der også virker på matricer fx

$$\operatorname{norm}\left(\begin{bmatrix}1 & 2 \\ 3 & 4 \\ 5 & 6\end{bmatrix}\right) \blacktriangleright \sqrt{91}$$

$$\sqrt{1^2+2^2+3^2+4^2+5^2+6^2} \blacktriangleright \sqrt{91}$$

Den virker selvfølgelig også på række-vektorer og søjle-vektorer, men den virker *ikke* på liste-vektorer

$$\operatorname{norm}(\{1,2,3\}) \blacktriangleright \text{Fejl: Ugyldig datatype}$$

hvor den som vist udløser en fejlmeddelelse. Man er derfor nødt til først at pakke liste-vektoren ind i en matrix med en krøllet parentes, før norm-kommandoen virker:

$\mathbf{u}:=\{1,2,3\} \blacktriangleright \{1,2,3\}$

$\mathrm{norm}(\{\mathbf{u}\}) \blacktriangleright \sqrt{14}$

På samme måde er det nemt at udregne determinanten ved at pakke vektorerne ind i en krøllet parentes:

$\mathbf{a}:=\{1,2\}:\mathbf{b}:=\{3,4\}:\det(\{\mathbf{a},\mathbf{b}\}) \blacktriangleright -2$

Det er til gengæld ret så krøllet, hvis det drejer sig om række-vektorer eller søjlevektorer, hvor man er nødt til at støtte sig til den såkaldte augment-kommando, fx

$\mathbf{c}:=\begin{bmatrix}1\\2\end{bmatrix}:\mathbf{d}:=\begin{bmatrix}3\\4\end{bmatrix}:\det(\mathrm{augment}(\mathbf{c},\mathbf{d})) \blacktriangleright -2$

Endnu værre ser det ud for 3-dimensionale vektorer:

$\mathbf{a}:=\begin{bmatrix}1\\2\\3\end{bmatrix} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \mathbf{b}:=\begin{bmatrix}4\\5\\6\end{bmatrix} \blacktriangleright \begin{bmatrix}4\\5\\6\end{bmatrix} \quad \mathbf{c}:=\begin{bmatrix}7\\8\\9\end{bmatrix} \blacktriangleright \begin{bmatrix}7\\8\\9\end{bmatrix}$

$\det(\mathrm{augment}(\mathrm{augment}(\mathbf{a},\mathbf{b}),\mathbf{c})) \blacktriangleright 0$

Konvertering af vektortyperne

Hvis man ønsker at konvertere en vektor fra en type til en anden er det i øvrigt forholdsvis nemt, som vist i det følgende skema:

Tip
Transponerings-tegnet $^{\mathrm{T}}$ findes i **tegnover-sigten** i venstre sidepanel.

Konvertering af de tre vektortyper	Liste-vektor $\mathbf{u}:=\{1,2,3\}$	Række-vektor $\mathbf{v}:=\begin{bmatrix}1&2&3\end{bmatrix}$	Søjle-vektor $\mathbf{w}:=\begin{bmatrix}1\\2\\3\end{bmatrix}$
Liste-vektor $\mathbf{u}:=\{1,2,3\}$		$\mathbf{v}:=\{\mathbf{u}\} \blacktriangleright \begin{bmatrix}1&2&3\end{bmatrix}$	$\mathbf{w}:=(\{\mathbf{u}\})^{\mathrm{T}} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix}$
Række-vektor $\mathbf{v}:=\begin{bmatrix}1&2&3\end{bmatrix}$	$\mathbf{u}:=\mathrm{mat}\blacktriangleright\mathrm{list}(\mathbf{v}) \blacktriangleright \{1,2,3\}$		$\mathbf{w}:=\mathbf{v}^{\mathrm{T}} \blacktriangleright \begin{bmatrix}1\\2\\3\end{bmatrix}$
Søjle-vektor $\mathbf{w}:=\begin{bmatrix}1\\2\\3\end{bmatrix}$	$\mathbf{u}:=\mathrm{mat}\blacktriangleright\mathrm{list}(\mathbf{w}) \blacktriangleright \{1,2,3\}$	$\mathbf{v}:=\mathbf{w}^{\mathrm{T}} \blacktriangleright \begin{bmatrix}1&2&3\end{bmatrix}$	

9 Differentialligninger

Ved hjælp af en række eksempler viser vi, hvordan man kan løse differentialligninger ved hjælp af Ti-nSpireCAS.

Der vises både symbolske og grafiske løsningsmetoder.

1. ordens differentialligninger

Statisk løsning af differentialligninger

Småkager bages ved 225°. Når de tages ud af ovnen, stilles de til afkøling i et 20° varmt rum. Med $y(t)$ betegnes småkagernes temperatur til tiden *t*.

I en model (Newtons afkølingslov) regner man med, at den hastighed $y'(t)$, hvormed temperaturen ændrer sig, er bestemt ved differentialligningen:

$$y' = -k \cdot (y-20),$$

hvor *k* er en positiv konstant.

Efter at småkagerne har været ude af ovnen i 1,0 minut, er deres temperatur 150°.

Løs differentialligningen, og bestem konstanten k.

Obs
Læg mærke til, at i differentialligninger skrives differentialkvotienten med symbolet '. Differentialligninger er det eneste sted i TI-Nspire CAS, hvor den afledede markeres med '.

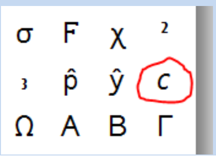
Til symbolsk løsning af differentialligninger benyttes kommandoen **deSolve**. Syntaksen for **desolve** er

deSolve(*differentialligning*, *uafhængig variabel*, *afhængig variabel*)

$\text{deSolve}(y'=-k\cdot(y-29),t,y) \blacktriangleright y=c1\cdot e^{-k\cdot t}+29$

$\text{deSolve}(y'=-k\cdot(y-20) \text{ and } y(0)=225,t,y) \blacktriangleright y=205\cdot e^{-k\cdot t}+20$

Tip
Det specielle kursiverede ***c***, der benyttes til konstanter i differentialligninger findes under **Tegn**



Først har vi ladet vi nSpireCAS løse differentialligningen generelt. Derved optræder der en *arbitrær* (vilkårlig) konstant ***c1*** i løsningen. Derefter er begyndelsesbetingelsen indsat. Nu bestemmer nSpireCAS så ***c1*** til at være 205.

I begge tilfælde mangler vi at få bestemt værdien af tallet *k*. Det hænger sammen med, at vi endnu ikke har brugt oplysningen $y(1,0)=150$.

Vi indsætter derfor $t=1,0$ og $y=150$ i den ufærdige løsning $y=205\cdot e^{-k\cdot t}+225$:

$\text{deSolve}(y'=-k\cdot(y-20) \text{ and } y(0)=225,t,y) \blacktriangleright y=205\cdot e^{-k\cdot t}+20$

$\text{solve}(y=205\cdot e^{k\cdot t}+20,k)|t=1. \text{ and } y=150 \blacktriangleright k=-0.455476$

$y=205\cdot e^{k\cdot t}+20|k=-0.455476 \blacktriangleright y=205\cdot(0.634146)^t+20$

Hermed har vi fundet løsningen (den partikulære løsning) til det givne differentialligningsproblem. Løsningen kan skrives på to måder

$$y=205\cdot e^{-0,455\cdot t}+225 \quad \text{eller} \quad y=205\cdot 0,634^t+225,$$

hvor *y* er temperaturen af småkagerne til tiden *t* minutter, efter at de kom ud af ovnen.

Løs differentialligningen $y' = 2x \cdot \mathrm{e}^{-y}$, hvor det er givet, at $y(0) = 1$.

$$\text{deSolve}\left(y'=2\cdot x\cdot \mathbf{e}^{-y} \text{ and } y(0)=1,x,y\right) \blacktriangleright \mathbf{e}^{y}-\mathbf{e}=x^2$$
$$\text{solve}\left(\mathbf{e}^{y}-\mathbf{e}=x^2,y\right) \blacktriangleright y=\ln\left(x^2+\mathbf{e}\right)$$

Som det ses ovenfor, løser **deSolve**-rutinen ikke differentialligningen til bunds, men giver i stedet det *implicitte* svar $\mathrm{e}^y - \mathrm{e} = x^2$. Den *eksplicitte* løsning $y = \ln(x^2 + \mathrm{e})$ findes så ved efterfølgende at bruge solve på det fremkomne implicitte svar.

I dette tilfælde er løsningen defineret for alle reelle tal x, men som vi skal se i det følgende eksempel, er der en grund til, at deSolve-rutinen er så forsigtig.

Det er den samme differentialligning, vi nu ser på, men begyndelsesbetingelsen er ændret.

Løs differentialligningen $y' = 2x \cdot \mathrm{e}^{-y}$, hvor det er givet, at $y(4) = 1$.

Vi går frem på samme måde som ovenfor og finder denne gang

$$\text{deSolve}\left(y'=2\cdot x\cdot \mathbf{e}^{-y} \text{ and } y(4)=1,x,y\right) \blacktriangleright \mathbf{e}^{y}-\mathbf{e}=x^2-16$$
$$\text{solve}\left(\mathbf{e}^{y}-\mathbf{e}=x^2-16,y\right) \blacktriangleright y=\ln\left(x^2+\mathbf{e}-16\right) \text{ and } x<-\sqrt{16-\mathbf{e}} \text{ or } y=\ln\left(x^2+\mathbf{e}-16\right) \text{ and } x>\sqrt{16-\mathbf{e}}$$

Vi får altså løsningen $y = \ln(x^2 + \mathrm{e} - 16)$, men TI-nSpireCAS foreslår to forskellige definitionsintervaller, enten $x < -\sqrt{16-\mathrm{e}}$, dvs. $x < -3{,}64$ eller $x > \sqrt{16-\mathrm{e}}$, dvs. $x > 3{,}64$.

Da x-værdien 4 skal ligge i definitionsintervallet, bliver løsningen

$$y = \ln(x^2 + \mathrm{e} - 16)\ ,\ x > \sqrt{16-\mathrm{e}}\ .$$

TI-nSpireCAS klarer også visse ikke-separable differentialligninger, men en gang imellem må programmet give op.

$$\text{deSolve}\left(y'=x+y,x,y\right) \blacktriangleright y=\mathbf{c7}\cdot \mathbf{e}^{x}-x-1$$
$$\text{deSolve}\left(y'=x+y^2,x,y\right) \blacktriangleright y'=x+y^2$$

Bemærk, at der i det sidste tilfælde ikke kommer nogen fejlmeddelelse. Programmet giver bare den givne differentialligning uændret tilbage igen. Man kan faktisk vise, at løsningen til denne differentialligning slet ikke kan udtrykkes ved hjælp af standardfunktioner, så TI-Nspire CAS er lovligt undskyldt ☺

Løsningsformlerne til såvel separable differentialligninger som lineære differentialligninger er indbygget i TI-Nspire CAS som man kan se på det følgende skærmklip

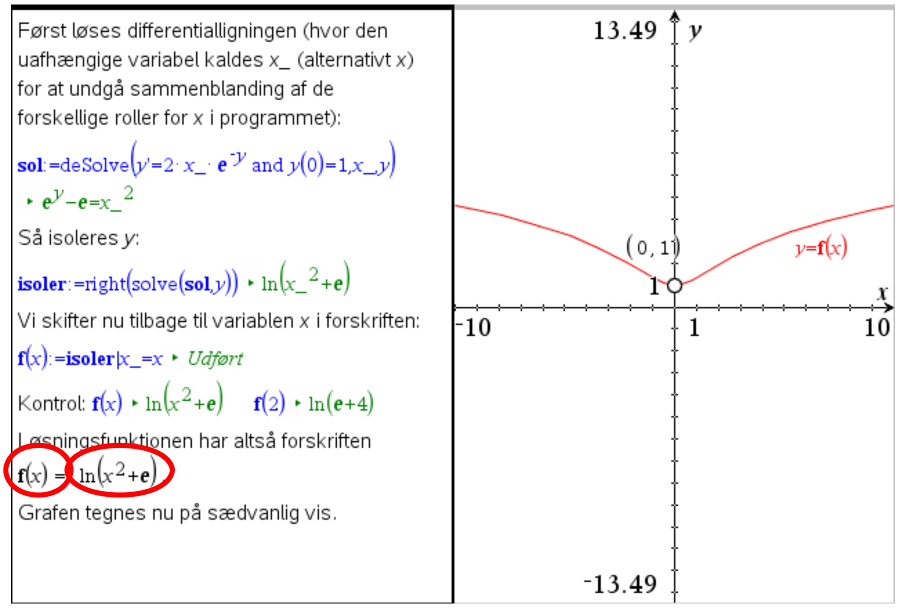
$$\text{deSolve}\left(y'=g(x)\cdot h(y),x,y\right) \blacktriangleright \int \frac{1}{h(y)}\,\mathrm{d}y=\int g(x)\,\mathrm{d}x+\mathbf{c8}$$
$$\text{deSolve}\left(y'=f(x)+g(x)\cdot y,x,y\right) \blacktriangleright y=\mathbf{e}^{\int g(x)\,\mathrm{d}x}\cdot\int\left(\mathbf{e}^{-\int g(x)\,\mathrm{d}x}\cdot f(x)\right)\mathrm{d}x+\mathbf{e}^{\int g(x)\,\mathrm{d}x}\cdot \mathbf{c9}$$

Interaktiv løsning af 1. ordens differentialligninger*

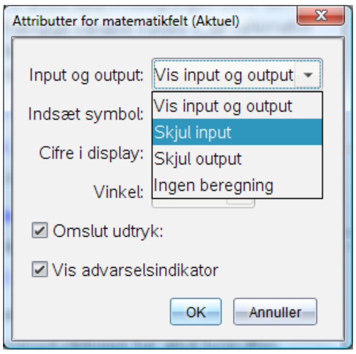
I det foregående har vi løst differentialligningerne enkelt og effektivt med brug af **deSolve**- og **Solve**-kommandoerne, idet vi undervejs i opgaven har *kopieret* de relevante udtryk og *indsat* dem i de efterfølgende beregninger, graftegninger osv. Men en sådan løsningsprocedure hænger *ikke* sammen interaktivt: Hvis der fx er en skrivefejl i den oprindelige differentialligning skal vi ind og gentage hele løsningsproceduren, da *kopiering* og *indsætning* ikke opdateres, når vi retter fejlen. Det samme er tilfældet hvis vi ønsker at genbruge en løsningsprocedure i anden sammenhæng: Hele løsningsproceduren skal gentages, i stedet for blot at rette differentialligningen til og lade resten opdatere af sig selv.

Meget kan altså vindes ved at konstruere løsningsproceduren interaktivt, men det er også teknisk mere udfordrende.Dertil kommer der et specielt problem med løsningfunktionen: Forskriften oplyses som et algebraisk udtryk og man skal være forsigtig, når dette udtryk konverteres til en egentlig funktion.

Løs differentialligningen $y' = 2x \cdot e^{-y}$, hvor det er givet, at $y(0) = 1$.
Tegn grafen for løsningsfunktionen til differentialligningen.



Det meste af fremgangsmåden fremgår direkte af teksten. Men i konklusionen har vi brugt to matematikbokse, en for $\boxed{f(x)=}$ og en for $\boxed{f(x)}$. Den sidste har vi så efterfølgende udregnet ved at taste **ENTER**, skifte farve til sort og efterfølgende valgt **Skjul input** under Attributter:



2. ordens differentialligninger

Vi ser på differentialligningen $y'' = -9y$. Dobbelt-mærket fås ved blot at taste enkelt-mærket to gange. I øvrigt er syntaksen den samme som for 1.ordens ligninger.

deSolve kan uden videre håndtere to former for begyndelsesbetingelser:

- løsning gennem et givet linjeelement (x_1, y_1, y_1')
- løsning gennem to givne punkter (x_1, y_1) og (x_2, y_2).

Obs
Husk, at vinkel-indstillingen skal være radian.

Generel løsning:

deSolve$(y''=-9\cdot y,x,y)$ ▸ $y=c3\cdot\cos(3\cdot x)+c4\cdot\sin(3\cdot x)$

Løsning gennem linjeelementet (0,2;3):

deSolve$(y''=-9\cdot y$ and $y(0)=2$ and $y'(0)=3,x,y)$ ▸ $y=2\cdot\cos(3\cdot x)+\sin(3\cdot x)$

Løsning gennem punkterne (0,1) og (π/2,3):

deSolve$\left(y''=-9\cdot y \text{ and } y(0)=1 \text{ and } y\left(\frac{\pi}{2}\right)=2,x,y\right)$ ▸ $y=\cos(3\cdot x)-2\cdot\sin(3\cdot x)$

NB! Følgende virker ikke:

Løsning gennem et punkt og med given hældning i et andet punkt:

deSolve$(y''=-9\cdot y$ and $y(0)=2$ and $y'(1)=3,x,y)$ ▸ Fejl: Argumentfejl

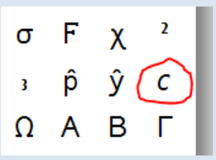
Løsning med given hældning i to punkter:

deSolve$(y''=-9\cdot y$ and $y'(0)=2$ and $y'(1)=3,x,y)$ ▸ Fejl: Argumentfejl

Opgaver som de to sidste i rammen ovenfor kan løses ved en kombination af **deSolve** og **solve**. Vi viser, hvordan den sidste kan løses.

Tip
fp med *p* for prime (dvs. mærke) er TI-Nspire CAS foretrukne betegnelse for den afledede funktion.

Tip
Det specielle kursiverede ***c***, der benyttes til konstanter i differentialligninger findes under **Tegn**



Vi bruger først deSolve med den ene hældning indsat:

deSolve$(y''=-9\cdot y$ and $y'(0)=2,x,y)$ ▸ $y=c5\cdot\cos(3\cdot x)+\frac{2\cdot\sin(3\cdot x)}{3}$

Derefter gemmes løsningen under navnet *f*(*x*):

$f(x):=c5\cdot\cos(3\cdot x)+\frac{2\cdot\sin(3\cdot x)}{3}$ ▸ *Udført*

Denne funktion differentieres:

$fp(x):=\frac{d}{dx}(f(x))$ ▸ *Udført*

Herefter indsættes den anden hældning:

$y'(1)=3$,

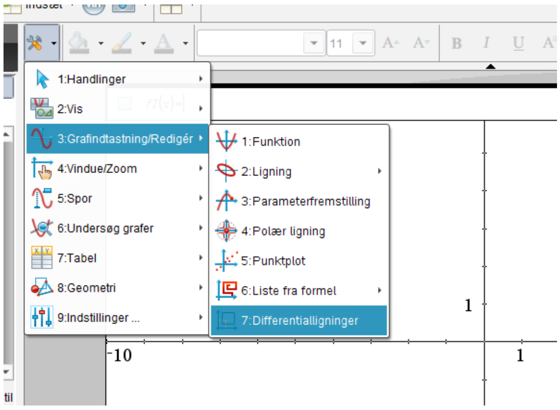
hvorefter $c5$ bestemmes:

solve$(fp(1)=3,c5)$ ▸ $c5=-11.763$

Altså er løsningen $y=-11.763\cdot\cos(3\cdot x)+\frac{2\cdot\sin(3\cdot x)}{3}$.

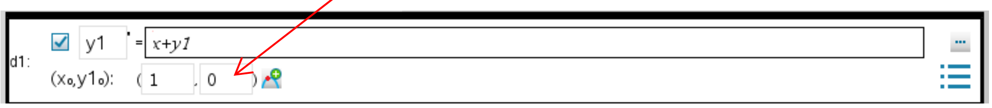
Grafisk løsning af differentialligninger

Vi vender tilbage til differentialligningen $y' = x + y$, som vi har løst symbolsk (side 2 i dette kapitel). Grafisk løsning af differentialligninger foregår naturligt nok i et **Graf**-vindue, hvor der skal skiftes indstilling:

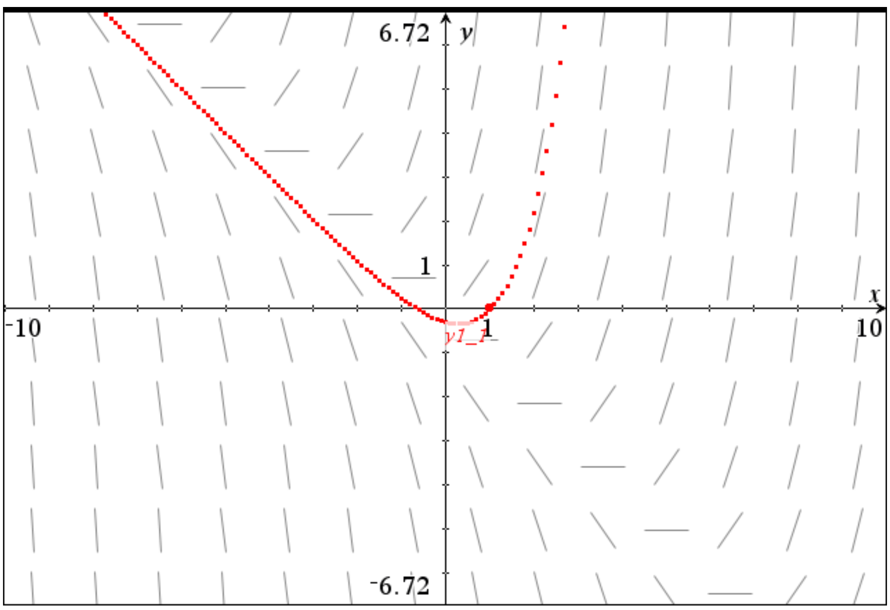


Derefter indtastes differentialligningen og begyndelsesbetingelsen. Bemærk, at de afhængige variable som standard hedder **y1**, **y2** osv. Men disse navne kan naturligvis udskiftes efter behov. Undgå dog x og y som betegnelser for afhængige variable ☺.

Indtastningslinjen ser nu således ud, idet de tomme felter til venstre er benyttet til at indtaste begyndelsesbetingelsen (1,0).



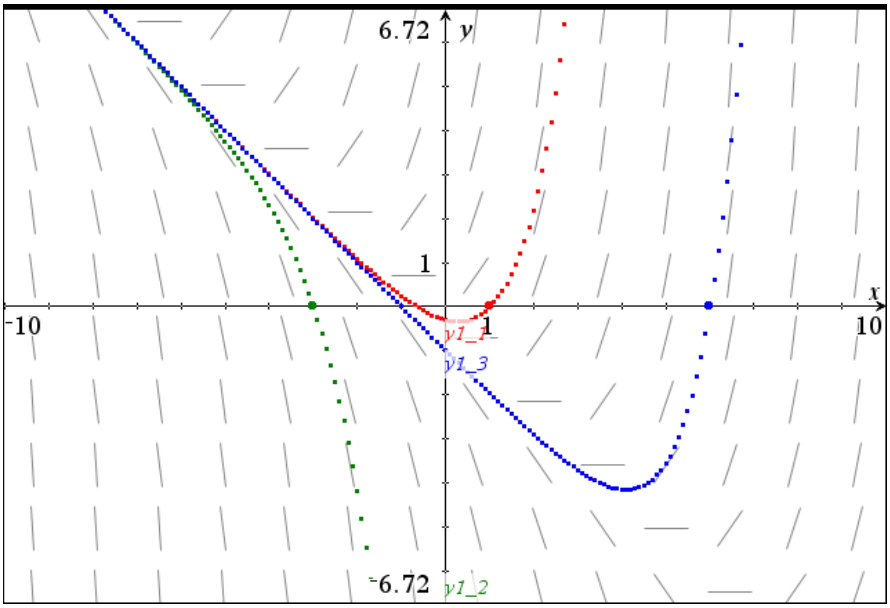
Når vi derefter trykker på **Enter**, sker der en hel masse:



Tip
Man kan trække i begyndelsespunktet for at se hvordan, løsningskurven afhænger af begyndelsesværdien.

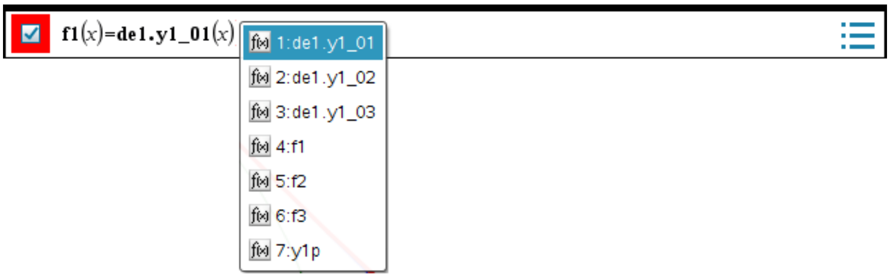
På figuren ses en stribe linjeelementer (standard er 14 i hver vandret række), som antyder en løsningskurves mulige forløb. Svarende til begyndelsesbetingelsen (1,0) (der er fremhævet som en særlig fed prik) itererer programmet sig ved Eulers metode frem langs linjeelementer, hvorved de røde punkter fremkommer.

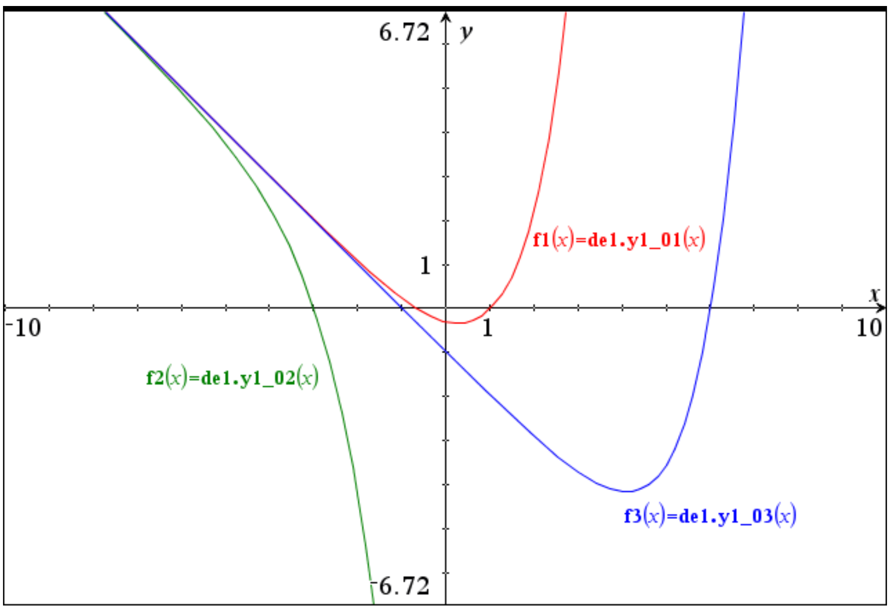
Man kan godt indtaste flere forskellige begyndelsesbetingelser. Klik på det grønne .
Figuren på næste side viser 3 løsninger.



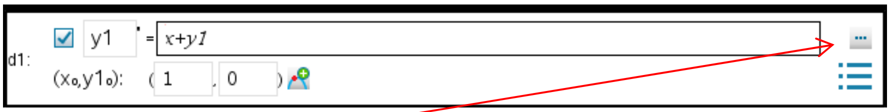
Efter at man har løst differentialligningerne grafisk, har man adgang til løsningerne opfattet som funktioner, så de kan tegnes i et normalt grafvindue.

Navnene hentes i variabel-registret og ses i øvrigt på nedenstående figur, hvor graferne er vist i præcis samme størrelsesforhold som ovenfor, men selvfølgelig kommer linjeelementerne *ikke* med her. Det gør de kun, hvis man skifter til sædvanlige funktioner i differentialligningsvinduet ☺





Vi vender tilbage til indtastningslinjen fra før.

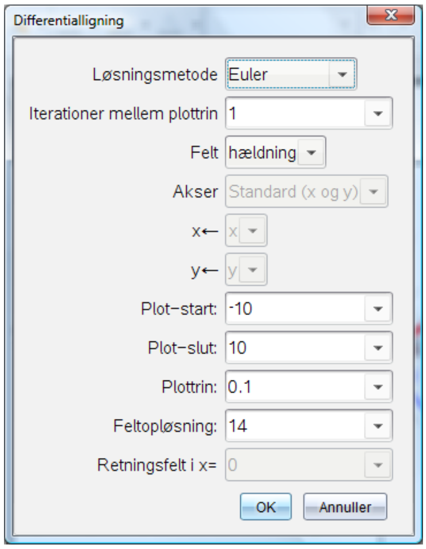


Ved klik på de tre prikker ⋯ øverst til højre på indtastningslinjen åbnes der en større dialogboks for differentialligningsvinduets parametre.

Obs
Parameterindstillingerne gælder for *alle* differentialligningsvinduer indenfor den samme opgave.

Obs
Det samlede antal løsningspunkter **plotinterval/plottrin** må *ikke* overstige 2500. Hvis fx x-intervallet går fra 0 til 1000 skal **Plottrin** sættes op til fx 1.

De viste standardindstillinger egner sig udmærket til de fleste1.ordens differentialligninger, men kan selvfølgelig ændres efter behov.
Forklaring på nogle af punkterne:
Iterationer mellem plottrin: Styrer hvor præcis den grafiske løsning skal være.
Felt: hældning svarer til, at der er tegnet linjeelementer.
Plot-start og **Plot-slut** svarer til vinduesgrænserne på x-aksen.
Plottrin: Angiver x-tilvæksten for løsningspunkterne.
Feltopløsning: Tallet 14 er antallet af linje-elementer, der tegnes i hver vandret række på figuren.

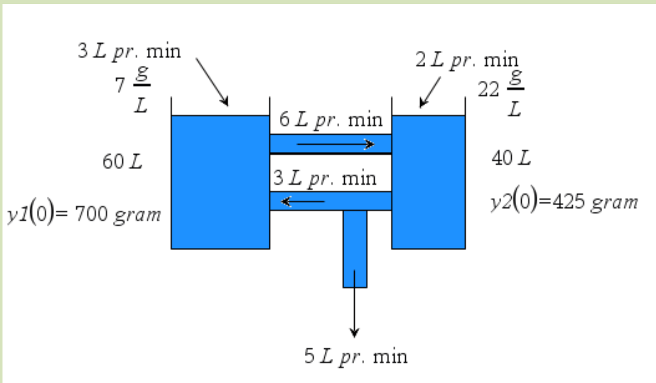


Koblede differentialligninger

Ved to koblede differentialligninger er der to *afhængige variable* $y1$ og $y2$. Koblingen består i, at hver af de to differentialligninger indeholder begge y-variable.

Vi ser på et eksempel med to forbundne kar.

Der strømmer saltvand ind i, mellem og ud af to kar.
60 L-beholderen indeholder til at begynde med 700 gram salt.
40 L-beholderen indeholder til at begynde med 425 gram salt.
Ved siden af pilene ses strømningshastigheden, målt i L/min.
For de to tilløb ses også saltkoncentrationen målt i gram/L.



Vi vil nu beskrive udviklingen af saltindholdet i de to kar.

Til tidspunktet t indeholder de henholdsvis $y1$ og $y2$ gram salt.

Vi ser først på 60 L beholderen:

60 L beholderen tilføres 3 L/min med koncentrationen 7 g/L, dvs. 21 g salt pr. minut udefra. Desuden strømmer der 6 L/min ud til den anden beholder med koncentrationen $\frac{y1}{60}$ g/L , dvs. $6\cdot\frac{y1}{60}=0{,}10\cdot y1$ gram salt pr. minut.

Endelig strømmer der 3 L/min ind fra den anden beholder med koncentrationen $\frac{y2}{40}$ g/L ,

dvs. $3 \cdot \frac{y2}{40} = 0{,}075 \cdot y2$ gram salt pr. minut.

Alt i alt fås differentialligningen for den 60 L-beholderen: $y1' = 21 - 0{,}10 \cdot y1 + 0{,}075 \cdot y2$.

På tilsvarende måde fås for 40 L-beholderen: $y2' = 44 + 0{,}10 \cdot y1 - 0{,}2 \cdot y2$.

Saltmængderne $y1$ og $y2$ i de to beholdere kan altså beskrives ved følgende lineære ligningssystem af *koblede differentialligninger*:

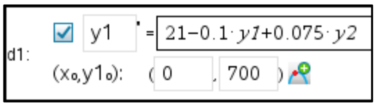
$$\begin{cases} y1' = 21 - 0{,}10 \cdot y1 + 0{,}075 \cdot y2 \\ y2' = 44 + 0{,}10 \cdot y1 - 0{,}2 \cdot y2 \end{cases}.$$

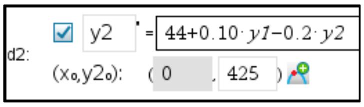
Obs
Husk at parameterindstillingerne gælder for *alle* differentialligningsvinduer indenfor den samme opgave. Det er derfor afgørende at du i det mindste starter på en ny opgave.

Vi skal åbne **Graf**-værkstedet til differentialligninger som vist tidligere i dette kapitel. Når man skal løse et sådant system grafisk, skal man gøre sig klart, at der i virkeligheden er tre variable på spil. Den *uafhængige variabel*, som kaldes x eller t, og de to *afhængige variable* $y1$ og $y2$. I koordinatsystem tegner vi et såkaldt *fasediagram*, der viser $y1$ på førsteaksen og $y2$ på andenaksen. Når vi skal vælge vinduet, er det dog nødvendigt at overveje alle tre variable.

De sædvanlige vinduesgrænser på figuren for x og y bestemmes af $y1$ og $y2$, men vi skal også tage stilling til værdien af **Plot-start** og **Plot-slut**, som svarer til det tidsinterval, vi vil betragte.

Vi viser først, hvordan de to differentialligninger indtastes, inkl. begyndelsesbetingelser.





I højre ende af indtastningslinjen åbner vi parameterdialogboksen med de tre prikker ... . Menuen, der åbner sig, ses til højre.

Som **Løsningsmetode** vælges Runge-Kutta.

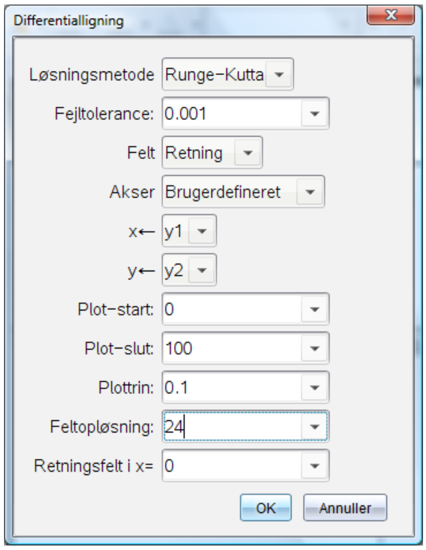
Fejltolerancen angiver hvor præcis vi ønsker at fastlægge løsningen.

Ved **Felt** vælges denne gang Retning. $y1$ og $y2$ anbringes på x- og y-aksen.

Som tidsinterval,dvs. **Plot-start** og **Plot-slut** vælges fra 0 til 100.

Vi lader **Plottrin** stå på 0.1 svarende til 1000 løsningspunkter, hvilket er under det maksimalt tilladte antal på 2500.

Feltopløsning sættes op til fx 25.

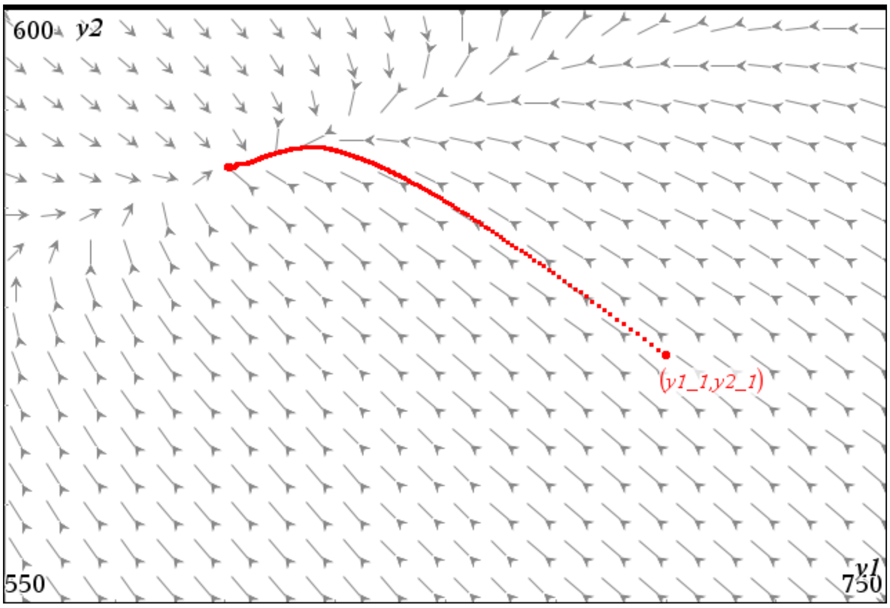


Tip
Runge-Kutta32 er en adaptiv Runge-Kutta metode til løsning af differentialligninger, hvor man løser den med en tredje-ordens og en anden-ordens metode samtidig. Steplængden tilpasses, så forskellen mellem de to løsninger holdes under fejltolerancen. **Runge-Kutta32** er derfor langt mere præcis end **Euler** med fast steplængde.

Valg af **Vinduesgrænser** for $y1$ og $y2$ sker først på selve grafen. Det viser sig at være hensigtsmæssigt at vælge 550 til 750 som x-grænser (dvs. $y1$-grænser) og 300 til 600 som y-grænser (dvs. $y2$-grænser). Vi skifter også aksebetegnelserne ud, så der står $y1$ og $y2$ på akserne.

Løsningen ses på næste side.

Tip
Hvis du vil se hvordan løsningspunktet bevæger sig langs løsningskurven kan du spore løsningen (hold højre piletast nede for at følge sporet).



På figuren ser vi begyndelsespunktet (700, 425) svarende til, at saltindholdet i 60 L-beholderen er 700 gram og i 40 L-beholderen er 425 gram til at begynde med. Man kan også se på de stroboskopiske prikker (afsat med lige store tidsrum) at løsningpunktet bevæger sig ret hurtigt til at begynde med for så at gå i stå, når løsningskurven nærmer sig slutpunktet /ligevægtspunktet.

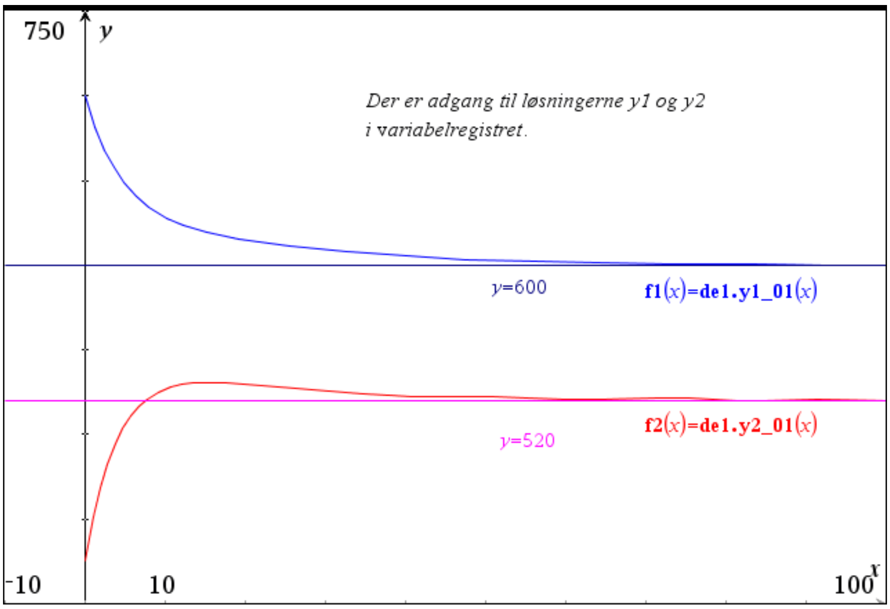
Slutpunktet på grafen er (600, 520), som svarer til, at saltindholdet i de to beholdere stabiliserer sig på 600 gram og 520 gram. De sidste værdier er løsningerne til ligningssystemet

$$\begin{cases} y1' = 21 - 0{,}10 \cdot y1 + 0{,}075 \cdot y2 \\ y2' = 44 + 0{,}10 \cdot y1 - 0{,}2 \cdot y2 \end{cases},$$

hvor der på venstre side i begge ligninger indsættes 0.

At saltindholdet i de to beholdere stabiliserer sig, kan også ses ved at tegne $y1$ og $y2$ som funktion af x i et sædvanligt grafvindue.

Tip
Den hurtigste måde at tilføje vandrette og lodrette linjer til et **Graf**-vindue er at indskrive ligningerne i tekstbokse, og dernæst trække tekstboksen ind på en af akserne.



Lotka-Volterra's rovdyr-byttedyr model

En berømt eksempel på koblede differentialligninger er rovdyr-byttedyr modellen, hvor $y1$ = antal kaniner (byttedyr) og $y2$ = antal ræve (rovdyr).

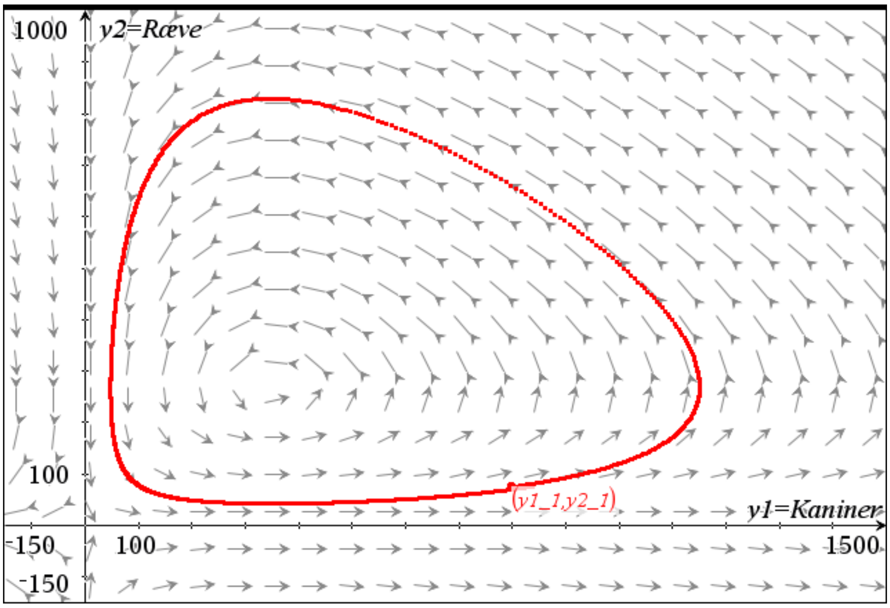
Modellen kan fx se således ud:

$$\begin{cases} y1' = 0,08 \cdot y1 - 0,0003 \cdot y1 \cdot y2 \\ y2' = 0,0002 \cdot y1 \cdot y2 - 0,07\, y2 \end{cases}.$$

Koefficienternes betydning:

0,08: kaninernes vækstrate (uden ræve)
0,0003: kaninernes dødsrate (med ræve) = rævenes succesrate
0,0002: rævenes vækstrate (med byttedyr)
0,07: rævenes dødsrate (uden byttedyr).

Med begyndelsesbetingelser 800 kaniner og 75 ræve fås nedenstående grafiske løsning.



Udviklingen går mod uret på denne graf. Begyndelsespunktet (800, 75) er markeret. Man ser, hvordan den voksende kaninbestand efterhånden fører til en voksende rævebestand, hvorved kaninbestanden falder. Men så falder rævebestanden, og kaninbestanden vokser igen osv.

2. ordens differentialligninger

Udsvinget u i et bestemt fjedrende system med gnidning eller luftmodstand kan beskrives ved differentialligningen

$$u'' + 0,125u' + u = 0$$

med begyndelsesbetingelserne $u(0)=2$ og $u'(0)=0$.

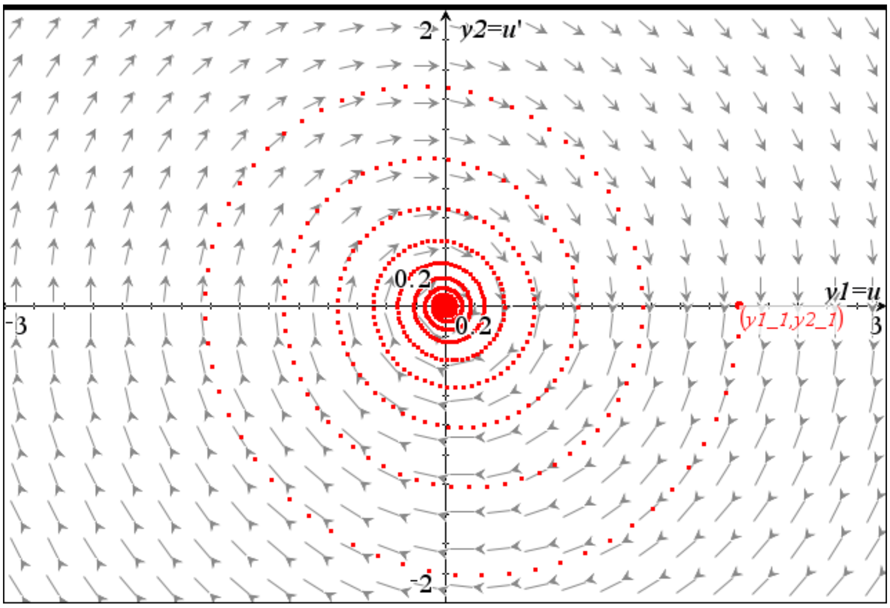
En sådan ligning kan løses med **deSolve**. Vi vil nu se, hvordan den løses grafisk.

Det gøres ved et lille trick, hvor differentialligningen omskrives til et system af to koblede differentialligninger ved at sætte $y1 = u$ og $y2 = u'$:

$$\begin{cases} y1' = y2 \\ y2' = -y1 - 0,125 \cdot y2 \end{cases}$$

med begyndelsesbetingelserne $y1(0) = 2$ og $y2(0) = 0$.

Løsningen til de koblede differentialligninger med Runge-Kutta-metoden ses på nedenstående diagram over $(u, u')=(y1, y2)$, hvor tiden x løber fra 0 til 100. Man ser, hvordan svingningerne efterhånden dør ud.

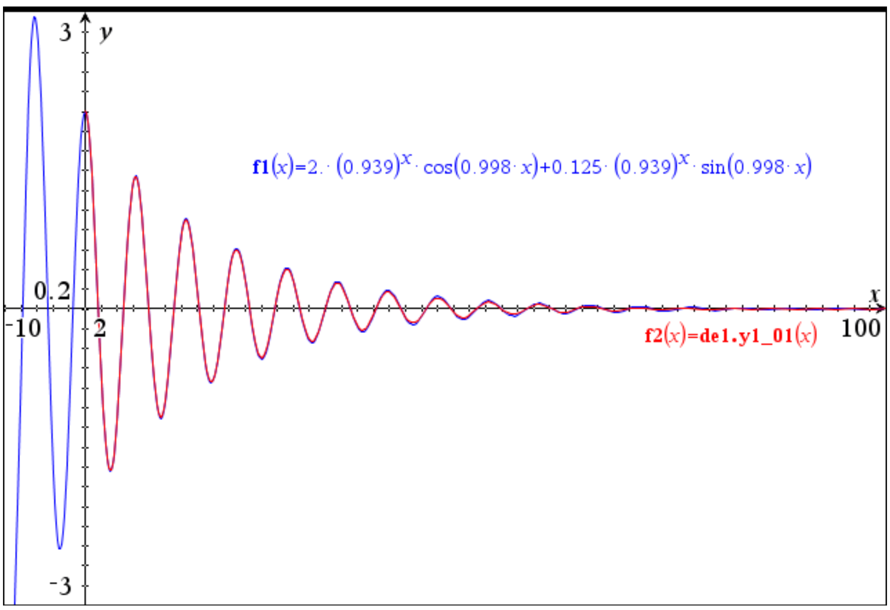


På nedenstående figur ses grafen for lønings-funktionen u fundet ved hjælp af **deSolve**.

$$\text{deSolve}\left(u''+0.125\cdot u'+u=0 \text{ and } u(0)=2 \text{ and } u'(0)=0,x,u\right)$$

$$u=2.\cdot(0.939413)^x\cdot\cos(0.998045\cdot x)+0.125245\cdot(0.939413)^x\cdot\sin(0.998045\cdot x)$$

Dertil kommer den numerisk grafiske løsning af differentialligningen overlejret den eksakte løsning, så man kan se hvor godt de passer sammen.



Dynamisk og interaktiv grafisk løsning af koblede differentialligninger*

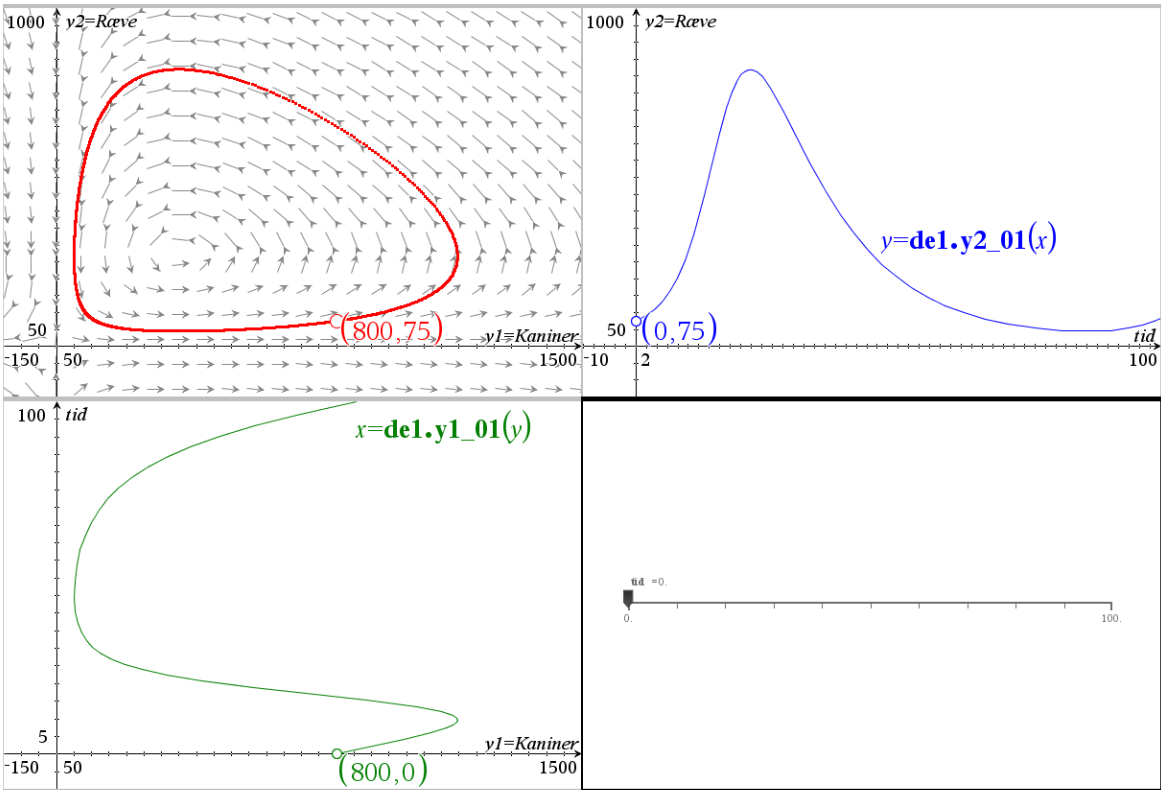
Som vi har set involverer koblede differentialligninger hele tre variable, en uafhængig variabel x og to afhængige variable $y1$ og $y2$. Det kan gøre det lidt sværere at gennemskue de forskellige sammenhænge. Man kan da kaste lys over betydningen af den grafiske løsning ved at oprette fire **Graf**-vinduer, der hænger indbyrdes sammen og dertil indføre en skyder for tiden x, så man kan trække i skyderen og se den tidslige udvikling på de tre grafer $(x,y1)$, $(x,y2)$ og $(y1,y2)$ samtidigt.

Tip
Et **faserumsplot** oprettes ved at sætte parametrene **Metode** til Runge Kutta og **Felt** til Retning. Endelig sættes **Plot-Start** og **Plot-Slut** til start- og slutværdi for tiden.

Vi bruger igen Lotka-Volterras rovdyr-byttedyr model som eksempel:

$$\begin{cases} y1' = 0.08\cdot y1 - 0.0003\cdot y1\cdot y2 & y1(0)=800 \\ y2' = 0.0002\cdot y1\cdot y2 - 0.07\,y2 & y2(0)=75 \end{cases}, \quad 0 \le x \le 100$$

Vi starter med at splitte i fire vinduer og opretter først faserumsplottet i øverste venstre hjørne. Derefter tilpasses det højre graf-vindue foroven til den samme lodrette akse som i faserumsplottet, mens den vandrette akse er tidsaksen. Tilsvarende tilpasses det venstre graf-vindue forneden til den samme vandrette akse som i faserumsplottet, mens den lodrette akse er tidsaksen (læg mærke til, at der er byttet om på den uafhængige og afhængige variabel i dette diagram!). Endelig har vi konstrueret en skyder for tiden i et geometrivindue i nederste højre vindue. Skydervariablen hedder **tid**.



For at tegne $(x,y2)$-grafen i øverste højre vindue kunne vi bare indtaste forskriften på sædvanlig vis. Men vi kunne også oprette en tekstboks $\boxed{y = de1.y2\_01(x)}$ og trække den ind på en koordinatakse.

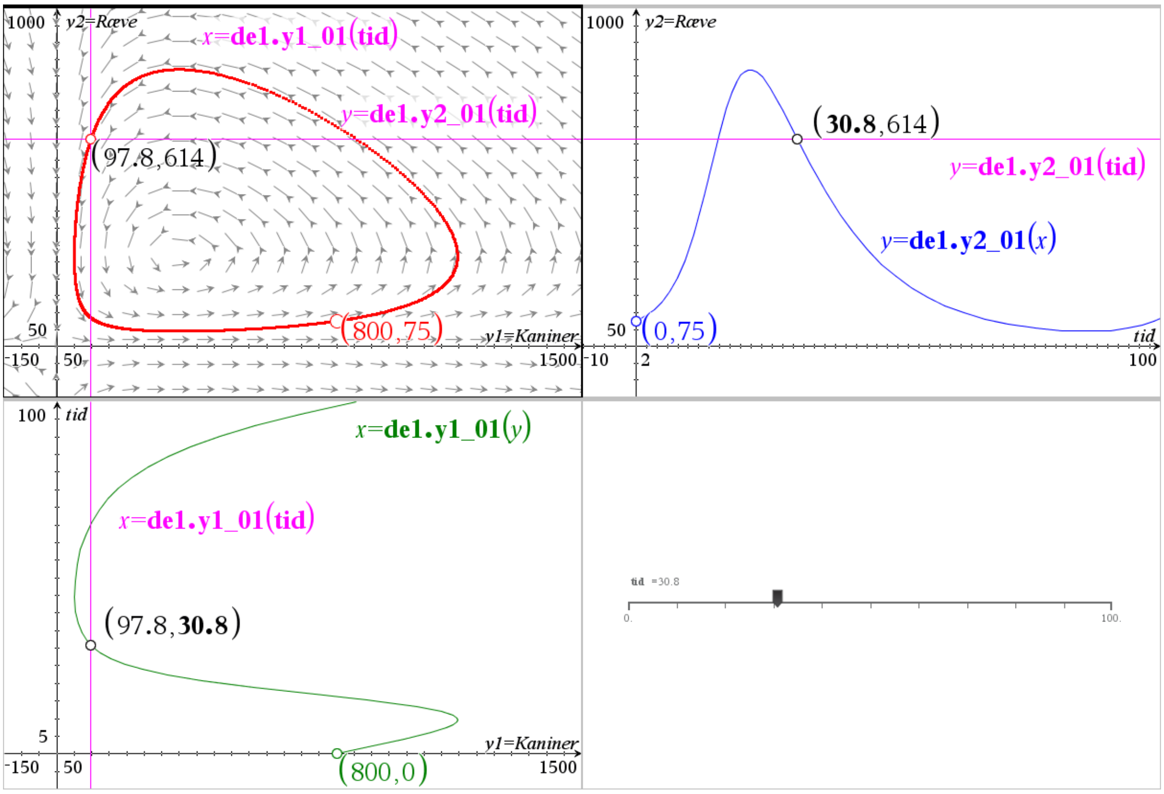
For at tegne $(y1,x)$-grafen i nederste venstre vindue er vi nødt til at bruge tekstboksmetoden, da der er byttet om på den uafhængige variabel og den afhængie variabel. Denne gang trækker vi derfor tekstboksen $\boxed{x = de1.y1\_01(y)}$ ind på en koordinatakse.

Så skal vi bare have afsat løsningspunkterne hørende til tidsparameteren ind på graferne.

Vi starter i øverste højre vindue: Her afsættes et punkt på grafen og førstekoordinaten kædes til variablen **tid**. Derefter afsættes en vandret linje ved hjælp af tekstboksen $\boxed{y = de1.y2\_01(tid)}$, der trækkes ind på en koordinatakse.

Vi fortsætter i nederste venstre vindue: Her afsættes et punkt på grafen og andenkoordinaten kædes til variablen **tid**. Derefter afsættes en lodret linje ved hjælp af tekstboksen $\boxed{x = de1.y1\_01(tid)}$, der trækkes ind på en koordinatakse.

Endelig skrives de samme to tekstbokse ind i det øverste venstre vindue (faserumsplottet) og de trækkes ind på en koordinatakse. Skæringspunktet mellem den vandrette og lodrette linje er da netop løsningspunktet i faserumsplottet.

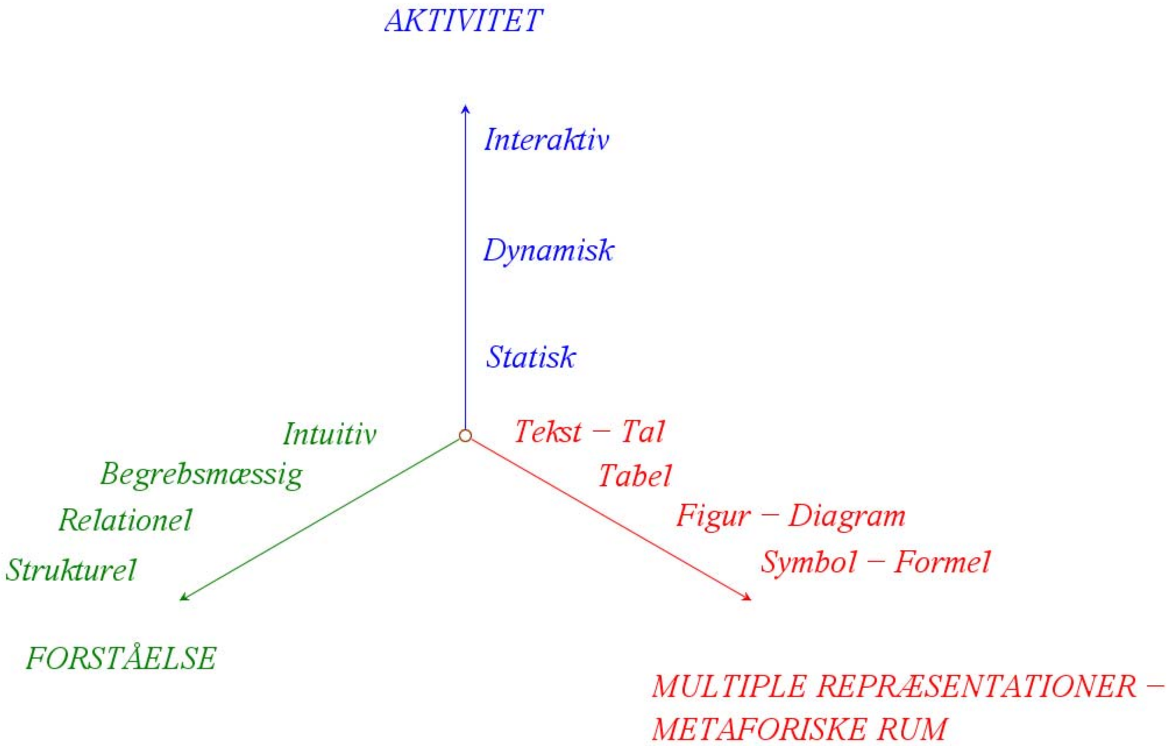


Trækker vi nu i skyderen for **tid** ser vi de sammenhørende løsningspunkter bevæge sig rundt på de tre grafer, der viser sammenhængene mellem de tre variable.

Vi har dermed fremstillet en såvel dynamisk som interaktiv illustration af løsningskurverne hørende til de koblede differentialligninger.

TI-nspire™ CAS

Et didaktisk funderet CAS-værktøj
Til matematikundervisningen



T³-Teachers Teaching with Technology™